Enhanced Data-Recalibration: Utilizing Validation Data to Mitigate Instance-Dependent Noise in Classification

Saeed Bakhshi Germi[1 [0000−0003−3048−220X]] and
Esa Rahtu[1 [0000−0001−8767−0864]]

Tampere University, Tampere, Finland
saeed.bakhshigermi@tuni.fi, esa.rahtu@tuni.fi

Abstract. This paper proposes a practical approach to deal with instance-dependent noise in classification. Supervised learning with noisy labels is one of the major research topics in the deep learning community. While old works typically assume class conditional and instance-independent noise, recent works provide theoretical and empirical proof to show that the noise in real-world cases is instance-dependent. Current state-of-the-art methods for dealing with instance-dependent noise focus on data-recalibrating strategies to iteratively correct labels while training the network. While some methods provide theoretical analysis to prove that each iteration results in a cleaner dataset and a better-performing network, the limiting assumptions and dependency on knowledge about noise for hyperparameter tuning often contrast their claims. The proposed method in this paper is a two-stage data-recalibration algorithm that utilizes validation data to correct noisy labels and refine the model iteratively. The algorithm works by training the network on the latest cleansed training Set to obtain better performance on a small, clean validation set while using the best performing model to cleanse the training set for the next iteration. The intuition behind the method is that a network with decent performance on the clean validation set can be utilized as an oracle network to generate less noisy labels for the training set. While there is no theoretical guarantee attached, the method's effectiveness is demonstrated with extensive experiments on synthetic and real-world benchmark datasets. The empirical evaluation suggests that the proposed method has a better performance compared to the current state-of-the-art works. The implementation is available at https://github.com/Sbakhshigermi/EDR.

Keywords: Label Noise · Classification · Data-Recalibration.

1 Introduction

Inexperienced workers, insufficient information about samples, confusing patterns, tiresome nature of the work, and other factors make the manual labeling of samples in a large dataset prone to errors and noisy labels [10, 29]. Unfortunately, deep learning algorithms have the potential to memorize these noisy

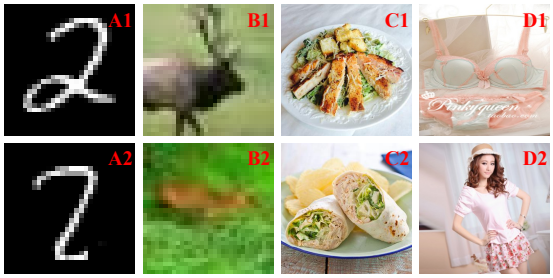

Fig. 1. Multiple samples of the same category in different datasets: (A) Number two in MNIST [15], (B) Deer in CIFAR-10 [14], (C) Caesar salad in Food-101N [16], and (D) Underwear in Clothing1M [34]. The images on the top row are more straightforward to label than the images on the bottom row.

labels, which leads to poor generalization and lower performance on clean test datasets [37]. Due to the importance of the topic in different sectors, such as safety-critical applications [2] and medical imaging [23], researchers have been developing methods to mitigate such label noise [10, 3, 27].

Most recent works assume the labels to be affected by a class-conditional noise (CCN) where the noise is instance-independent [20]. This type of noise can be estimated [13] or mitigated by adding extra loss terms in the model [4]. However, Chen utilized visual examples and mathematical analysis to prove that the label noise in a real-world dataset (Clothing1M [34]) is actually instance-dependent. To better understand why this is the case, take a look at Figure 1. As seen in this figure, two samples of the same category have different complexity of labeling, which suggests that the label noise is instance-dependent.

With the previous assumption of CCN proven wrong, a new mathematical foundation for mitigation methods had to be developed. Therefore, researchers started defining variations of instance-dependent noise (IDN) patterns to represent synthetic noise and propose mitigation approaches based on them. One of the effective strategies used in the state-of-the-art methods is the iterative data-recalibration [18]. These methods use the predictions of a network trained over noisy samples to select and correct samples iteratively.

While recent works on IDN provided theoretical analysis to prove the convergence of their models to an oracle Bayes classifier [5, 38], the limiting assumptions in their theories cannot be met in practical implementation, as shown by their empirical findings. Due to these limitations, this paper will focus on empirical experiments on synthetic and real-world datasets to showcase the effectiveness of the proposed method.

This paper proposes an enhanced data-recalibration algorithm that corrects labels affected by instance-dependent noise by utilizing validation set. On each iteration, the proposed method trains a model with the cleansed data from the last iteration to achieve higher performance on a small, clean validation set. Then, the best-performing model is chosen to correct labels in the training set

based on the model's confidence for the next iteration. The intuition is that better performance on the clean validation set means a better prediction of training labels than the previous iteration.

The main difference between the proposed method and previous works is utilizing a clean validation set to influence the training stage to help the network approach an oracle model that can predict ground truth labels. Small, clean validation sets can be easily obtained with computer-assisted tools [1]. While previous works often use the validation set as a selector of the final model for accuracy reports, none utilize it any further to the best of the authors' knowledge. The main contributions of this paper are:

- Proposing a practical data-recalibration algorithm that utilizes easy-to-gather clean validation set to enhance the performance over the existing state-of-the-art methods.
- Providing empirical evaluation with extensive experiments on both synthetic and real-world datasets to show the effectiveness of the proposed method.

The rest of the paper is structured as follows. Section 2 covers the related works. Next, Section 3 explains the proposed method in detail. After that, Section 4 deals with the experiments and the empirical evaluation to show the effectiveness of the proposed method. Finally, Section 5 concludes the work.

2 Related Works

Menon provided one of the major theoretical frameworks for IDN in binary problems. This framework provided the basis to construct a loss function with specific criteria to mitigate IDN. While the work was necessary at the time, the method is not extensible to deep neural networks [21]. Chen provided mathematical proof that the label noise in a large real-world dataset called Clothing1M [34] follows the IDN pattern. They proposed a method of generating IDN patterns by averaging the predictions of an oracle classifier over the training session to find complex samples and flip their labels. The mitigation method provided also relies on averaging the predictions of a network, with the intuition that the network can find a soft representation of labels that are closer to ground truth over time. While this work provided essential information about IDN, the mitigation method is cost-heavy with low performance compared to other works [7].

Zhang defined a new family of noise called poly-margin diminishing (PMD). This new noise family follows the same intuition that data points near the decision boundary are more challenging to classify, thus more prone to noise. Based on the previously stated reasons for label noise, this definition seems realistic. To mitigate this family of noise, they proposed an iterative correction method that corrects the labels based on the network confidence over the training set in each iteration. While the work provided theories to prove the effectiveness, their hyperparameter settings and assumption violation in implementing the method contradict their idea [38].

Several state-of-the-art methods managed to reach high performance on real-world benchmarks. Tan combined a supervised and an unsupervised network and co-teach them with the help of an encoder to maximize the agreement between the networks in latent space [28]. Wu utilized the spatial topology of data in the latent space of the network iteratively to collect clean labels and refine the network further [32]. Zhu focused on the second-order approach to estimate covariance terms for IDN with peer loss function [19] and defined a new loss function to change the problem to CCN [40]. Xia eliminated the need for anchor points in estimating the noise transition matrix [33]. Han described a two-stage algorithm where the trained network is used to select multiple class prototypes to represent the characteristics of the data better and correct the noisy labels [11]. Lee focused on reducing human supervision by introducing a method that required a small clean training set to extract the information about label noise [16]. Li divides the training data into labeled clean and unlabeled noisy samples to utilize semi-supervised learning techniques by training two networks and correcting more labels over each iteration [17]. Other methods such as PENCIL [36], ILFC [5], CORES^2 [8], Meta-Weight-Net [24], estimation of transition matrix [35], and JoCoR [31] are also noteworthy.

3 Proposed Method

In this section, we present the details of our proposed method. The proposed method alternates between training the network to find the best performance on the clean validation set and correcting the noisy labels based on confidence scores from the top-performing network. Before the proposed algorithm starts the process, we prepare a deep neural network by training it for a few epochs with a high learning rate, which allows the network to reach a reasonable confidence level without overfitting to noise [37].

3.1 Preliminaries

Let $\mathcal{X}$ be the feature space, $\mathcal{L}$ be the label space, $(x, y), (x, \tilde{y}) \in \mathcal{X} \times \mathcal{L}$ be a clean and a noisy sample respectively, $D = \{(x_i, y_i)\}_{i=1}^{n}$ be a dataset, $f^t(x) = (C_1, \ldots, C_k)$ be a classifier at the t-th iteration of the algorithm, where C_i is the confidence score of the network for the i-th class (output of softmax layer in this paper), and k is the total number of classes. Finally, let S^t be the performance of the classifier over clean validation set at t-th iteration of the algorithm.

3.2 Iterative Label Correction Method

The overall algorithm is summarized in Algorithm 1. In practice, we use an average of confidence scores from several top-performing networks. Since there is no guarantee of improving the network on every iteration, there might be a random instance where the trained network arbitrarily achieves a high performance score. Averaging multiple confidence scores mitigates the effect of these random

Algorithm 1: Enhanced Data-Recalibration

Require: Initial training set $\widetilde{D}^0_{train} = \left\{\left(x_i, \tilde{y}^0_i\right)\right\}^n_{i=1}$, Initial classifier f^0, threshold value θ, Number of epochs T, Validation set $D_{valid} = \{(x_i, y_i)\}^m_{i=1}$

for $t \in 1, \ldots, T$ **do**
  Train f^{t-1} on $\widetilde{D}^{t-1}_{train}$ to get f^t and get the performance score S^t
  Compare S^t to previous scores $\{S^i\}^{t-1}_{i=1}$ to find best performing classifier f^B
  for $(x, \tilde{y}) \in \widetilde{D}^{t-1}_{train}$ **do**
    Get the confidence scores $(C_1, \ldots, C_k)$ of f^B on x
    Find the best confidence score C_M and the noisy confidence score C_N
    Calculate $Gap = |\log(C_M) - \log(C_N)|$
    if $Gap \geq \theta$ **then**
      Set new label $\tilde{y}^t = M$
    else
      Keep old label $\tilde{y}^t = \tilde{y}^{t-1}$
    end if
  end for
  if $\forall i \in [1, \ldots, n]$, $\tilde{y}^t_n = \tilde{y}^{t-1}_n$ **then**
    Decrease θ by a small amount
  end if
end for

return Best trained network f^B

encounters as they do not introduce a bias towards any class. Moreover, the top-performing networks are selected from a range of recently trained networks to ensure that the network is not stuck in a loop. In the following subsections, we will describe what happens in the t-th iteration of the algorithm:

3.3 Stage One

In this stage, the algorithm starts training the network for one epoch with the labels acquired from the previous iteration. In other terms, the network from the previous iteration f^{t-1} is trained on the training set with labels generated in the previous iteration $\widetilde{D}^{t-1}_{train} = \left\{\left(x_i, \tilde{y}^{t-1}_i\right)\right\}^n_{i=1}$ to obtain the new network f^t. Then, the performance of the network is evaluated to obtain the top-performing network for the next stage. It is done by evaluating the new network f^t on the clean validation set $D_{valid} = \{(x_i, y_i)\}^m_{i=1}$ to get its performance score S^t. Then, this performance score S^t is compared to all previous scores $\{S^i\}^{t-1}_{i=1}$ to find the best-performing network $\left\{f^B \mid \forall i \leq t : S^B \geq S^i\right\}$.

3.4 Stage Two

In this stage, the algorithm starts collecting the confidence scores of the chosen network on the training set. It is done by predicting the confidence scores $(C_1, \ldots, C_k)$ of the best-performing network f^B for each sample in training set

from the previous iteration $(x, \tilde{y}) \in \widetilde{D}^{t-1}_{train}$. Then, the confidence scores are evaluated to decide the labels for the next iteration. For each sample in the dataset $(x, \tilde{y}) \in \widetilde{D}^{t-1}_{train}$, the highest confidence score $\{C_M \mid \forall i \leq k : C_M \geq C_i\}$ and the confidence score for the noisy label $C_{N=\tilde{y}}$ are considered. If the difference of logarithms between them is greater than a threshold $|\log(C_M) - \log(C_N)| \geq \theta$, then the sample is selected for correction. The intuition behind the process is that a noticeable gap between the prediction of the best-performing network and the current label suggests the label is noisy. After that, the labels for the next iteration are generated. It is done by swapping the label of the selected samples to the prediction of the best-performing network $\tilde{y}^t_{sel} = M$ while keeping the labels of other samples the same as before $\tilde{y}^t_{rest} = \tilde{y}^{t-1}$. Finally, the threshold value is evaluated and reduced if the algorithm cannot select samples anymore. By initializing a high threshold value and lowering it in small steps, the best-performing network gains more trust from the algorithm gradually, which prevents confirmation bias to some degree.

4 Experiments & Evaluation

4.1 Synthetic Datasets

For proof of concept, the public datasets CIFAR-10 and CIFAR-100 [14] are chosen for synthetic experiments. Both datasets contain 50,000 training and 10,000 testing samples over ten categories. In the case of CIFAR-100, each category is further divided into ten subclasses. As argued by the previous works [7, 5, 38], a realistic noise does not uniformly affect all data space points. The most common solution among previous works to generate reliable IDN is to find challenging samples and then flip their label from the most confident category to the second most confident category. A challenging sample is typically located at the edges of the decision boundary and results in a low network confidence score. Such samples can be found by training an oracle network and selecting the low confidence samples [5] or averaging the network's confidence over the training period and selecting the confusing samples [7]. To generate reliable and comparable IDN, we follow the definition for the PMD noise family [38].

Let $\aleph_{C_1,C_2}(x) = \mathbb{P}[\tilde{y} = C_2 \mid y = C_1, x]$ be the probability of corrupting the label of a sample from the most confident class C_1 to the second-most confident class C_2, and $f^*(x)$ be an oracle classifier trained on clean samples. The three types of IDN used in our experiments are defined as in Equation 1.

$$
\begin{aligned}
\aleph^{\mathrm{I}}_{C_1,C_2}(x) &= \frac{1}{2} - \frac{1}{2}\left[f^*_{C_1}(x) - f^*_{C_2}(x)\right]^2 \\
\aleph^{\mathrm{II}}_{C_1,C_2}(x) &= 1 - \left[f^*_{C_1}(x) - f^*_{C_2}(x)\right]^3 \\
\aleph^{\mathrm{III}}_{C_1,C_2}(x) &= 1 - \frac{1}{3}\left[f^*_{C_1}(x) - f^*_{C_2}(x)\right]^3 \\
&\quad - \frac{1}{3}\left[f^*_{C_1}(x) - f^*_{C_2}(x)\right]^2 - \frac{1}{3}\left[f^*_{C_1}(x) - f^*_{C_2}(x)\right]
\end{aligned}
\tag{1}
$$

For the sake of completion, we also include the most common CCN noise types in our experiments: uniform and asymmetrical [22]. Let $\beth_{C_1,C_2} = \mathbb{P}[\tilde{y} = C_2 \mid y = C_1]$ be the probability of corrupting the label of a sample from class C_1 to class C_2, $\mathcal{R}$ be the noise rate and k be the total number of classes. The two types of CCN used in our experiments are defined as in Equation 2.

$$\begin{aligned} \beth_{C_1,C_2}^{\text{Uniform}} &= \begin{cases} \frac{\mathcal{R}}{k-1} & C_1 \neq C_2 \\ 1-\mathcal{R} & C_1 = C_2 \end{cases} \\ \beth_{C_1,C_2}^{\text{Asymmetrical}} &= \begin{cases} \mathcal{R} & C_1 \neq C_2 \\ 1-\mathcal{R} & C_1 = C_2 \end{cases} \end{aligned} \tag{2}$$

The ResNet-34 [12] is used for synthetic experiments. All models are trained from scratch for 180 epochs with a batch size of 128 images. Stochastic gradient descent is used as the optimizer with a momentum value equal to 9×10^{-1} and a weight decay rate of 5×10^{-4}. The learning rate is initialized as 1×10^{-2} and gets divided by 2 after 40 and 80 epochs. Standard data augmentations are applied: random horizontal flip, 32×32 random crop after padding 4 pixels, and standard normalizing with mean = (0.4914, 0.4822, 0.4465), std = (0.2023, 0.1994, 0.2010). In each experiment, 10% of the clean training data is reserved as the validation set. Each experiment is repeated 5 times to report the mean and standard deviation for final accuracy. The initial value for θ in Algorithm 1 is set to 7×10^{-1} with a decrement step of 1×10^{-1}. The algorithm averages 5 top-performing networks from the last 30 epochs on each iteration.

Table 1. Final accuracy on the CIFAR datasets for different IDN patterns and rates.

Dataset	Noise Info	SL[30]	LRT[39]	PLC[38]	**Ours**
CIFAR-10	$\aleph^{\text{I}}_{35\%}$	79.76 ± 0.7	80.98 ± 0.8	82.80 ± 0.3	**83.60 ± 0.3**
	$\aleph^{\text{I}}_{70\%}$	36.29 ± 0.7	41.52 ± 4.5	42.74 ± 2.1	**46.47 ± 1.1**
	$\aleph^{\text{II}}_{35\%}$	77.92 ± 0.9	80.74 ± 0.3	81.54 ± 0.5	**83.41 ± 0.3**
	$\aleph^{\text{II}}_{70\%}$	41.11 ± 1.9	44.67 ± 3.9	46.04 ± 2.2	**46.24 ± 0.9**
	$\aleph^{\text{III}}_{35\%}$	78.81 ± 0.3	81.08 ± 0.4	81.50 ± 0.5	**83.16 ± 0.3**
	$\aleph^{\text{III}}_{70\%}$	38.49 ± 1.5	44.47 ± 1.2	45.05 ± 1.1	**46.33 ± 1.1**
CIFAR-100	$\aleph^{\text{I}}_{35\%}$	55.20 ± 0.3	56.74 ± 0.3	60.01 ± 0.4	**63.85 ± 0.3**
	$\aleph^{\text{I}}_{70\%}$	40.02 ± 0.9	45.29 ± 0.4	45.92 ± 0.6	**46.38 ± 0.3**
	$\aleph^{\text{II}}_{35\%}$	56.10 ± 0.7	57.25 ± 0.7	63.68 ± 0.3	**63.91 ± 0.3**
	$\aleph^{\text{II}}_{70\%}$	38.45 ± 0.6	43.71 ± 0.5	45.03 ± 0.5	**46.63 ± 0.2**
	$\aleph^{\text{III}}_{35\%}$	56.04 ± 0.7	56.57 ± 0.3	63.68 ± 0.3	**63.92 ± 0.4**
	$\aleph^{\text{III}}_{70\%}$	39.94 ± 0.8	44.41 ± 0.2	44.45 ± 0.6	**46.22 ± 0.2**

Table 1 holds the results of testing the proposed method on synthetic data affected by three different IDN patterns with 35% and 70% noise rates. The performance of baseline methods is obtained from [38]. As shown in this table,

our method outperforms the alternatives in all cases. Judging by the numbers, some alternative approaches have a high standard deviation rate, indicating possible instability of that method.

Table 2. Final accuracy on the CIFAR datasets for different combinations of Noise.

Dataset	Noise Info	SL[30]	LRT[39]	PLC[38]	**Ours**
CIFAR-10	$\aleph^{\mathrm{I}}_{35\%} + \beth^{\mathrm{Uniform}}_{30\%}$	77.79 ± 0.5	75.97 ± 0.3	79.04 ± 0.5	**80.94 ± 0.2**
	$\aleph^{\mathrm{I}}_{35\%} + \beth^{\mathrm{Asymmetrical}}_{30\%}$	77.14 ± 0.7	76.96 ± 0.5	78.31 ± 0.4	**79.93 ± 0.5**
	$\aleph^{\mathrm{II}}_{35\%} + \beth^{\mathrm{Uniform}}_{30\%}$	75.08 ± 0.5	75.94 ± 0.6	80.08 ± 0.4	**81.07 ± 0.2**
	$\aleph^{\mathrm{II}}_{35\%} + \beth^{\mathrm{Asymmetrical}}_{30\%}$	75.43 ± 0.4	77.03 ± 0.6	77.63 ± 0.3	**79.90 ± 0.5**
	$\aleph^{\mathrm{III}}_{35\%} + \beth^{\mathrm{Uniform}}_{30\%}$	76.22 ± 0.1	75.66 ± 0.6	80.06 ± 0.5	**80.54 ± 0.3**
	$\aleph^{\mathrm{III}}_{35\%} + \beth^{\mathrm{Asymmetrical}}_{30\%}$	76.09 ± 0.1	77.19 ± 0.7	77.54 ± 0.7	**79.54 ± 0.5**
CIFAR-100	$\aleph^{\mathrm{I}}_{35\%} + \beth^{\mathrm{Uniform}}_{30\%}$	51.34 ± 0.6	45.66 ± 1.6	60.09 ± 0.2	**61.46 ± 0.4**
	$\aleph^{\mathrm{I}}_{35\%} + \beth^{\mathrm{Asymmetrical}}_{30\%}$	50.18 ± 1.0	52.04 ± 0.2	56.40 ± 0.3	**59.94 ± 0.4**
	$\aleph^{\mathrm{II}}_{35\%} + \beth^{\mathrm{Uniform}}_{30\%}$	50.58 ± 0.3	43.86 ± 1.3	60.01 ± 0.6	**61.16 ± 0.3**
	$\aleph^{\mathrm{II}}_{35\%} + \beth^{\mathrm{Asymmetrical}}_{30\%}$	49.46 ± 0.2	52.11 ± 0.5	**61.43 ± 0.3**	59.34 ± 0.5
	$\aleph^{\mathrm{III}}_{35\%} + \beth^{\mathrm{Uniform}}_{30\%}$	50.18 ± 0.5	42.79 ± 1.8	60.14 ± 1.0	**61.82 ± 0.3**
	$\aleph^{\mathrm{III}}_{35\%} + \beth^{\mathrm{Asymmetrical}}_{30\%}$	48.15 ± 0.9	50.31 ± 0.4	54.56 ± 1.1	**59.76 ± 0.5**

Table 2 holds the results of testing the proposed method on synthetic data simultaneously affected by IDN and CCN patterns. The final noise rate is typically lower than the sum of two individual noise rates due to overlaps in selected samples. As shown in this table, our method still outperforms the alternatives in almost all cases.

4.2 Real-world Datasets

To evaluate the performance of the proposed method on real-world cases, three commonly used datasets were chosen for testing:

ANIMAL-10N [26] – This dataset contains 50,000 training and 5,000 testing samples over ten categories. According to the creators of the dataset, the estimated noise rate is about 8%. Following the authors' work, we chose VGG-19 [25] with a batch normalization for this experiment. The model is trained from scratch for 180 epochs with a batch size of 128 images. Stochastic gradient descent is used as the optimizer with a weight decay rate of 1×10^{-3}. The learning rate is initialized as 1×10^{-1} and gets divided by 5 after 50 and 75 epochs. Standard data augmentations are applied: random horizontal flip and standard normalizing with mean = (0.485, 0.456, 0.406), std = (0.229, 0.224, 0.225). 10% of the training data is manually labeled with the help of [1] and reserved as the validation set. The initial value for θ in Algorithm 1 is set to 7×10^{-1} with a decrement step of 1×10^{-1}. The algorithm averages 10 top-performing networks from the last 30 epochs on each iteration. Table 3 holds the results of testing

the proposed method on the ANIMAL-10N dataset. The performance of baseline methods is obtained from their respective papers. As seen in this table, the proposed method outperforms the alternatives.

Table 3. Final accuracy on the Animal-10N and Food-101N datasets.

Dataset	Method	Accuracy	Dataset	Method	Accuracy
Animal-10N	SELFIE [26]	79.40	Food-101N	DeepSelf [11]	79.40
	Co-Learning [28]	82.95		PLC [38]	83.40
	PLC [38]	83.40		Ours	86.34
	Ours	**84.47**		**Co-Learning** [28]	**87.57**

Food-101N [16] – This dataset contains 310,000 training samples and utilizes the 25,000 testing samples provided by the Food-101 dataset [6] over 101 categories. According to the creators of the dataset, the estimated noise rate is about 10%. Following the authors' work, we chose ResNet-50 with pre-trained weights on ImageNet [9] for this experiment. The model is fine-tuned for 30 epochs with a batch size of 32 images. Stochastic gradient descent is used as the optimizer with a weight decay rate of 1×10^{-3}. The learning rate is initialized as 5×10^{-3} and gets divided by 10 after 10 and 20 epochs. Standard data augmentations are applied: random horizontal flip, 224×224 random crop, and standard normalizing with mean = (0.485, 0.456, 0.406), std = (0.229, 0.224, 0.225). 14% of the labels are verified by the creators of the dataset to be used as the validation set. The initial value for θ in Algorithm 1 is set to 9×10^{-1} with a decrement step of 1×10^{-1}. The algorithm averages 4 top-performing networks from the last 8 epochs on each iteration. Table 3 holds the results of testing the proposed method on the Food-101N dataset. The performance of baseline methods is obtained from their respective papers. This table shows that the proposed method outperforms most of the alternatives but gets beaten by Co-Learning [28].

Clothing1M [34] [16] – This dataset contains 1,000,000 samples over 14 categories, out of which 50,000 training, 14,000 validation, and 10,000 testing samples are verified by the creators of the dataset. Following the previous works [17, 16, 32], the clean training data is discarded. We chose ResNet-50 with pre-trained weights on ImageNet for this experiment. The model is fine-tuned for 20 epochs with a batch size of 32 images. Stochastic gradient descent is used as the optimizer with a momentum value equal to 9×10^{-1} and a weight decay rate of 5×10^{-4}. The learning rate is initialized as 1×10^{-3} and gets divided by 10 after 5 and 10 epochs. Standard data augmentations are applied: random horizontal flip, 224×224 random crop, and standard normalizing with mean = (0.485, 0.456, 0.406), std = (0.229, 0.224, 0.225). The verified validation data is used as the validation set. The initial value for θ in Algorithm 1 is set to 3×10^{-1} with a decrement step of 1×10^{-1}. The algorithm averages 4 top-performing networks

from the last 8 epochs on each iteration. Table 4 holds the results of testing the proposed method on the Clothing1M dataset. The performance of baseline methods is obtained from their respective papers. As seen in this table, the proposed method outperforms the alternatives.

Table 4. Final accuracy on the Clothing1M dataset.

Method	Accuracy
CAL [40]	74.17
Reweight [33]	74.18
DeepSelf [11]	74.45
CleanNet [16]	74.69
DivideMix [17]	74.76
Ours	**75.11**

5 Conclusion

This paper proposes a practical iterative label correction method that utilizes clean validation sets to achieve better performance when dealing with instance-dependent noise. The effectiveness of the proposed method is shown with empirical experiments on both synthetic and real-world benchmark datasets. The proposed method outperformed the current state-of-the-art methods in these experiments. The findings suggest that the proposed method's intuition might be correct, and utilizing a clean validation set in iterative label correction methods is helpful.

References

1. Adhikari, B., Huttunen, H.: Iterative bounding box annotation for object detection. In: 25th International Conference on Pattern Recognition (ICPR). pp. 4040–4046 (2021). https://doi.org/10.1109/ICPR48806.2021.9412956
2. Adhikari, B., Peltomäki, J., Bakhshi Germi, S., Rahtu, E., Huttunen, H.: Effect of label noise on robustness of deep neural network object detectors. In: Computer Safety, Reliability, and Security. SAFECOMP 2021 Workshops. pp. 239–250 (2021)
3. Algan, G., Ulusoy, I.: Image classification with deep learning in the presence of noisy labels: A survey. Knowledge-Based Systems **215** (2021). https://doi.org/10.1016/j.knosys.2021.106771
4. Arazo, E., Ortego, D., Albert, P., O'Connor, N., Mcguinness, K.: Unsupervised label noise modeling and loss correction. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 312–321 (2019)
5. Berthon, A., Han, B., Niu, G., Liu, T., Sugiyama, M.: Confidence scores make instance-dependent label-noise learning possible. In: Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 825–836 (2021)

6. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: Computer Vision – ECCV. pp. 446–461. Proceedings of Machine Learning Research (2014)
7. Chen, P., Ye, J., Chen, G., Zhao, J., Heng, P.A.: Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. Proceedings of the AAAI Conference on Artificial Intelligence **35**(13), 11442–11450 (2021)
8. Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., Liu, Y.: Learning with instance-dependent label noise: A sample sieve approach (2021)
9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
10. Frenay, B., Verleysen, M.: Classification in the presence of label noise: A survey. IEEE Transactions on Neural Networks and Learning Systems **25**(5), 845–869 (2014). https://doi.org/10.1109/TNNLS.2013.2292894
11. Han, J., Luo, P., Wang, X.: Deep self-learning from noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
13. Hendrycks, D., Lee, K., Mazeika, M.: Using pre-training can improve model robustness and uncertainty. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 2712–2721 (2019)
14. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
15. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
16. Lee, K.H., He, X., Zhang, L., Yang, L.: Cleannet: Transfer learning for scalable image classifier training with label noise. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
17. Li, J., Socher, R., Hoi, S.C.H.: Dividemix: Learning with noisy labels as semi-supervised learning (2020)
18. Li, J., Wong, Y., Zhao, Q., Kankanhalli, M.S.: Learning to learn from noisy labeled data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
19. Liu, Y., Guo, H.: Peer loss functions: Learning from noisy labels without knowing noise rates. In: Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 6226–6236 (2020)
20. Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., Bailey, J.: Normalized loss functions for deep learning with noisy labels. In: Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 6543–6553 (2020)
21. Menon, A.K., van Rooyen, B., Natarajan, N.: Learning from binary labels with instance-dependent noise. Machine Learning **107**(8-10), 1561–1595 (2018). https://doi.org/10.1007/s10994-018-5715-3
22. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
23. Shi, J., Wu, J.: Distilling effective supervision for robust medical image segmentation with noisy labels (2021)

24. Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., Meng, D.: Meta-weight-net: Learning an explicit mapping for sample weighting (2019)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
26. Song, H., Kim, M., Lee, J.G.: SELFIE: Refurbishing unclean samples for robust deep learning. In: Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 97, pp. 5907–5915 (2019)
27. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: A survey (2021)
28. Tan, C., Xia, J., Wu, L., Li, S.Z.: Co-learning: Learning from noisy labels with self-supervision. In: Proceedings of the 29th ACM International Conference on Multimedia. p. 1405–1413 (2019)
29. Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.: Learning from noisy large-scale datasets with minimal supervision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6575–6583 (2017). https://doi.org/10.1109/CVPR.2017.696
30. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J.: Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
31. Wei, H., Feng, L., Chen, X., An, B.: Combating noisy labels by agreement: A joint training method with co-regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
32. Wu, P., Zheng, S., Goswami, M., Metaxas, D., Chen, C.: A topological filter for learning with label noise (2020)
33. Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., Sugiyama, M.: Are anchor points really indispensable in label-noise learning? In: Advances in Neural Information Processing Systems. vol. 32 (2021)
34. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
35. Yang, S., Yang, E., Han, B., Liu, Y., Xu, M., Niu, G., Liu, T.: Estimating instance-dependent label-noise transition matrix using dnns (2021)
36. Yi, K., Wu, J.: Probabilistic end-to-end noise correction for learning with noisy labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
37. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Communications of the ACM **64**(3), 107–115 (2021). https://doi.org/10.1145/3446776
38. Zhang, Y., Zheng, S., Wu, P., Goswami, M., Chen, C.: Learning with feature-dependent label noise: A progressive approach (2021)
39. Zheng, S., Wu, P., Goswami, A., Goswami, M., Metaxas, D., Chen, C.: Error-bounded correction of noisy labels. In: Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 11447–11457 (2020)
40. Zhu, Z., Liu, T., Liu, Y.: A second-order approach to learning with instance-dependent label noise. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10113–10123 (2021)