Coupled-View Deep Classifier Learning from Multiple Noisy Annotators

Shikun Li,[1,2] Shiming Ge,[1*] Yingying Hua,[1,2] Chunhui Zhang,[1,2]
Hao Wen,[3] Tengfei Liu,[4] Weiqiang Wang[4]
[1]Institute of Information Engineering, Chinese Academy of Sciences, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, China
[3]CloudWalk Technology Co., Ltd, China [4]Ant Financial Services Group
{lishikun, geshiming, huayingying, zhangchunhui}@iie.ac.cn,
wenhao@cloudwalk.cn, {aaron.ltf, weiqiang.wwq}@antfin.com

Abstract

Typically, learning a deep classifier from massive cleanly annotated instances is effective but impractical in many real-world scenarios. An alternative is collecting and aggregating multiple noisy annotations for each instance to train the classifier. Inspired by that, this paper proposes to learn deep classifier from multiple noisy annotators via a coupled-view learning approach, where the learning view from data is represented by deep neural networks for data classification and the learning view from labels is described by a Naive Bayes classifier for label aggregation. Such coupled-view learning is converted to a supervised learning problem under the mutual supervision of the aggregated and predicted labels, and can be solved via alternate optimization to update labels and refine the classifiers. To alleviate the propagation of incorrect labels, small-loss metric is proposed to select reliable instances in both views. A co-teaching strategy with class-weighted loss is further leveraged in the deep classifier learning, which uses two networks with different learning abilities to teach each other, and the diverse errors introduced by noisy labels can be filtered out by peer networks. By these strategies, our approach can finally learn a robust data classifier which less overfits to label noise. Experimental results on synthetic and real data demonstrate the effectiveness and robustness of the proposed approach.

1 Introduction

With the availability of a cleanly annotated large-scale dataset, recent deep learning has proven success in various classification tasks. For example, the ResNet model (He et al. 2016) has achieved 6.43% error rate in CIFAR10 object classification task (Krizhevsky 2009), while the ArcFace model (Deng et al. 2019) achieves a surpassing human-level accuracy of 99.82% in recognizing faces on the LFW benchmark (Huang et al. 2007). Generally, such success arises from deep networks' powerful capacity in extracting knowledge from massive labeled data in a supervised learning manner. However, the clean annotations for supervised learning are very difficult to collect in many real-world scenarios, e.g., video surveillance in the wild (Sun et al. 2018)

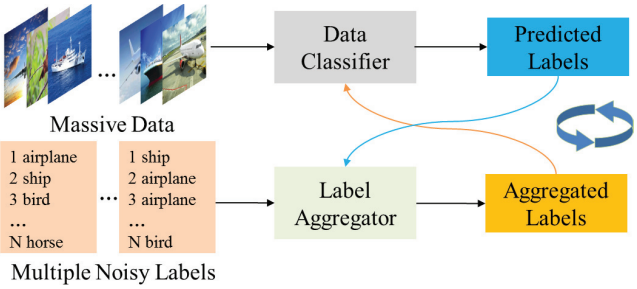

Figure 1: The coupled-view perspective of our learning approach. It alternately learns a data classifier and a label aggregator under the mutual supervision of the aggregated and predicted labels that are updated during training.

and medical data analysis (Amy 2018). Recently, the crowdsourcing approaches like Amazon Mechanical Turk (AMT) (Buhrmester, Kwang, and Gosling 2011) have been established as a cost-effective solution to annotate large collections of data. This manner provides a feasible way to create massive data with single or multiple noisy labels.

For learning from single noisy labels, some approaches (Salla et al. 2018; Liu and Tao 2016) tried to model noise confusion matrix or select the clean labels. They generally improve accuracy by learning from single noisy labels or aggregated labels. Recently, the co-teaching approach proposed by (Han et al. 2018) has showed good ability in selecting clean labels by using two networks with the same structure to update each other's training instances, leading to superior performance. Generally, these methods provide some effective ways to reduce the overfitting to label noises.

Learning from multiple noisy labels provides an intuitive way to lessen label noises. As noisy labels are relatively easy to acquire from multiple annotators (e.g., different devices, persons or models), the common practice is to collect multiple weak labels for an instance and then aggregate them to get a more reliable label. Toward this end, it is necessary to explore an effective solution that can address a key challenge: *how to aggregate multiple noisy annotators to facilitate deep classifier learning?* A simple approach is regarding the results of majority voting (Cheru-

*Shiming Ge is the corresponding author.

bin 2019) as the groundtruth labels. But this practice neglects the different characteristics of annotators, and the multiple annotators may obey various noise distributions, leading to uneven label quality. More advanced practices, like expectation maximization (EM) algorithm (Bekker and Goldberger 2016) and its variants (Huang and Chen 2017; Khetan, Anandkumar, and Lipton 2018), from the viewpoint of probability, jointly estimate the unknown biases of the annotators and network parameters. However, these algorithms are unstable and often doesn't converge to the optimal results or even get worse after a number of rounds.

To address the challenge, from the perspective of coupled-view learning, we propose to regard EM algorithm's joint estimation problem as a problem of mutual learning between two learning views: the learning view from data and the learning view from labels, as shown in Fig.1. The learning is performed alternatively to optimize both learning views: 1) the learning view from data aims to train a deep classifier with the supervision of aggregated labels from another view, and 2) the learning view from labels models the biases of annotators, then aggregates weak labels from them, with the help of the prediction of the deep classifier (known as predicted labels). In this way, the learning from multiple noisy annotators is transformed into a supervised learning problem, and next we are faced with the critical issue: *how to facilitate the learning converging to the good and stable results?* To make the learning convergent, the performance of both learning views should be improved progressively in each round. We experimentally find that the learning performance is affected by two main factors: the propagation of incorrect labels and the class-imbalance of correct labels. To address the first factor, we adopt small-loss metric to select cleaner instances in both views and further a co-teaching strategy in the learning view from data to mitigate random error bias introduced by initialization of deep networks. The second factor arises from the instance selection and thus we introduce a dynamic class-weighted loss in the learning view from data, and set a class-balanced prior distribution in another view. By these strategies, our approach has a more stable convergence and reduces the overfitting to wrong labels, which is a critical bottleneck in previous methods.

The main contributions of this paper are summarized as three folds: 1) We propose a coupled-view learning approach to learn deep classifiers from multiple noisy annotators, which outperform the other state-of-arts; 2) We propose small-loss metric to select reliable instances in both learning views and co-teaching strategy with a dynamic class-weighted loss to further reduce the effect of label noises in neural network, which facilitates the good and stable convergence of deep classifier learning; and 3) We conduct comprehensive experiments on synthetic and real data to demonstrate the effectiveness and robustness of the proposed approach, which may be helpful for developing deep classifiers in other real-world scenarios.

2 Related Works

The approach we proposed in this paper aims to learning classifier from multiple noisy labels with a coupled-view learning method. Therefore, we briefly review related works from two aspects, including learning with noisy labels and multi-view learning methods.

2.1 Learning with Noisy Labels

In order to learn from data with noisy labels, one direction of some recent works is modeling label noises or reducing the effect of noisy labels. Some methods focus on estimating the noise transition matrix or make probabilistic model for noises. For example, Bekker and Goldberger (2016) adopted EM algorithm to take turns to estimate true labels and noise transition. Goldberger and Ben-Reuven (2017) added an additional softmax layer after the true output layer to model the noise transition matrix. By the contrast, Liu and Tao (2016) proposed using surrogate loss for label noises and proved that any surrogate loss function can be used for classification with noisy labels by using importance reweighting. Yu et al. (2018) provided a method to modify traditional loss and extends standard neural network classifiers to learn with biased labels. Xia et al. (2019) presented a risk-consistent estimator and employed this estimator to tune the transition matrix. Another line is estimating true labels and then learning with them. PENCIL (Yi and Wu 2019) employed similar idea and proposed continuously updating the label distribution through backpropagation. There are also other attempts to refine the noisy labels. By using a small clear dataset and label relations in knowledge graph as assistance, Li, Yang, and Song (2017) proposed a unified distillation framework to hedge the risk of learning from noisy labels. These methods are generally effective at low noise level but most of them may suffer from errors when having heavy label noises. Recently, a new insight is selecting reliable noisy labels. Jiang et al. (2018) pretrained an guide network for selecting clean instances to update the learning network. Han et al. (2018) proposed co-teaching that leverages two networks' different learning abilities to filter out different types of errors introduced by noisy labels. Cheng et al. (2019) introduced the concept of distilled examples and propose a learning algorithm robust to label noise. When label noise is heavy, selective methods outperform most of other methods as the classifier will not fit in the dropped incorrect labels.

When having multiple noisy labels, a naive approach is aggregating the labels with majority voting. MBEM (Khetan, Anandkumar, and Lipton 2018), an improved EM algorithm, proposed combining the learning from labels and estimated workers' quality for learning. Rodrigues and Pereira (2018) proposed adding the crowd layer to the output of common network, and the layer adjusts the gradients coming from the labels of annotators. Guan et al. (2018) learned different models for every annotator, and the whole output is weighted integration of multiple models' output. A key challenging issue of these approaches is how to perform effective label aggregation to facilitate deep classifier and make it converge to the good and stable results.

2.2 Multi-view Learning

Multi-view learning is mainly applied for unsupervised and semi-supervised learning. The approaches have several styles such as co-training, multiple kernel learning, sub-

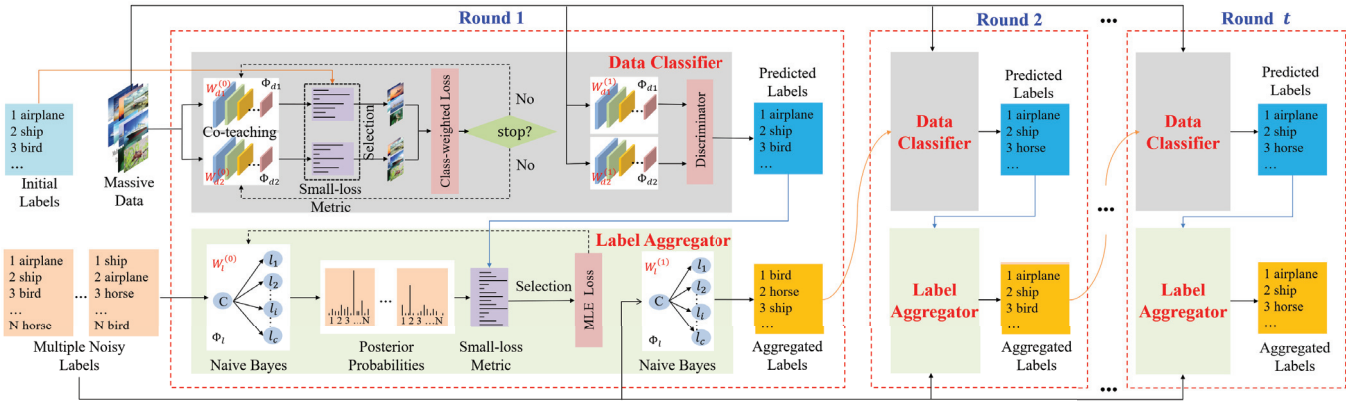

Figure 2: The whole framework of coupled-view learning. Several schemes including small-loss metric, co-teaching strategy and class-weighted loss are used to achieve the good and stable convergence in both learning views.

space learning and so on. Here we introduce co-training approaches which inspire our approach.

Co-training (Mitchell and Blum 1998) is one of the earliest approaches for multi-view learning, in which learners are trained alternately on two distinct views with confident labels for the unlabeled data. According to the measure way of label confidence, co-training has a series of variants, such as democratic co-learning (Goldman and Zhou 2000), cotrace (Zhang and Zhou 2011), tri-training (Zhou and Li 2005) and self-paced co-training (Ma et al. 2017). Wang and Zhou (2017) show that if the classifiers in two sufficient views have large diversity, co-training algorithms can improve performance. But if the view is insufficient, the optimal classifier will mistakenly classify some examples. The larger the insufficiency, the worse the performance of the optimal classifier (Wang and Zhou 2017).

Note that, although our problem setting is weakly-supervised learning, as both views in our method relabel instances in every round, we can see it as two-view models label unlabeled data in semi-supervised learning. That is to say, our method and co-training both exchange useful information by labeling instances. Hence, some theoretical analysis of co-training may be suited to our method. If our two views are conditionally independent and have heavy difference, which means label noise and data is independent when the class is known, we believe that the views can help to improve each other on certain conditions and it's demonstrated by our experiments. But due to insufficiency of both views, the propagation of incorrect labels may worse performance and our method proposes small-loss metric and co-teaching strategy to reduce it.

3 Our Approach

3.1 Problem Formulation

Our purpose refers to exploit an effective way to aggregate multiple weak labels to facilitate the learning of deep classifier from massive data. It needs to address two main issues: 1) *reliable aggregation from noisy labels* and 2) *robust learning from massive data*. Toward this end, our couple-view learning approach achieves that in a unified manner by collaborating two learning views (see Fig. 2): 1) the learning view from labels $\phi_l\left(\boldsymbol{y};\mathbf{w}_l\right)$ aggregates a m-dimensional noisy label vector $\boldsymbol{y}$ from fixed m annotators to generate an estimated distribution, which is used to update an aggregated label $y^l \in \{1,2,...,c\}$, and 2) the learning view from data $\phi_d\left(\boldsymbol{x};\mathbf{w}_d\right)$ is a deep classifier that takes an instance $\boldsymbol{x}$ (an image, a signal or a feature) as the input and outputs a predicted distribution, which can be converted to a predicted label $y^d \in \{1,2,...,c\}$. Here, $\mathbf{w}_d$ and $\mathbf{w}_l$ are the learned parameters of the classifiers ϕ_d and ϕ_l respectively, c is the size of label space or the class number. Thus, the objective of coupled-view learning is to learn the classifier parameters $\{\mathbf{w}_d,\mathbf{w}_l\}$ from given dataset $\mathbf{x}=\{\boldsymbol{x}_i\}_{i=1}^n$ and multiple noisy label set $\mathbf{y}=\{\boldsymbol{y}_i\}_{i=1}^n$, where n is the number of training instances and $\boldsymbol{y}_i=(y_{i1},y_{i2},...,y_{im})$ is a m-d noisy vector.

To achieve that, with the help of the predicted labels $\boldsymbol{y}^d=\{y_i^d\}_{i=1}^n$ and the aggregated labels $\boldsymbol{y}^l=\{y_i^l\}_{i=1}^n$ as two pseudo label sets, we transform this weakly-supervised learning problem into a mutual supervised learning one:

$$\min_{\mathbf{w}_d,\mathbf{w}_l} \ell_d\left(\mathbf{x},\boldsymbol{y}^l;\mathbf{w}_d\right)+\ell_l\left(\mathbf{y},\boldsymbol{y}^d;\mathbf{w}_l\right), \tag{1}$$

where $\ell_d(.)$ and $\ell_l(.)$ are the loss functions for training ϕ_d and ϕ_l, respectively. After initialization and pretraining, the problem can be solved with alternate optimization under the supervision of the predicted labels and the aggregated labels that are updated alternately during training. This coupled-view manner provides an simple and general way to make the label aggregator and the data classifier exchange knowledge with each other. Now, the focused issue is converted to the following: how to help the mutual learning converging to the good and stable results. To this end, we propose several strategies in both learning views that will be described in detail.

3.2 Learning View from Labels

Base model. With the assumption that each annotator's label noise is random and independent, we propose a Naive

Bayes classifier with noise confusion matrices as the base model, which is similar to (Albarqouni et al. 2016; Khetan, Anandkumar, and Lipton 2018). Based on Bayesian formula, when the noise confusion matrix $\boldsymbol{\pi}$ and the class distribution $\boldsymbol{q}$ are known, $\phi_l(\boldsymbol{y}_i; \mathbf{w}_l = (\boldsymbol{\pi}, \boldsymbol{q}))_k = p[y_i^t = k|y_i; \boldsymbol{\pi}, \boldsymbol{q}]$, the posterior probability of the true label of ith sample for kth class is calculated as Eq. (2).

$$\phi_l(\boldsymbol{y}_i; \boldsymbol{\pi}, \boldsymbol{q})_k = \frac{q_k \prod_{j=1}^{m} (\sum_{s=1}^{c} \mathbb{I}[y_{ij} = s] \pi_{ks}^{(j)})}{\sum_{k'=1}^{c} (q_{k'} \prod_{j=1}^{m} (\sum_{s=1}^{c} \mathbb{I}[y_{ij} = s] \pi_{k's}^{(j)}))}, \tag{2}$$

where y_{ij} is the jth weak label of the ith instance; $\mathbb{I}[.]$ is the indicator function which takes 1 if the identity index is true and 0 otherwise. $\pi_{ks}^{(j)}$ is the probability of misclassifying the kth class into the sth class for the jth annotator; q_k is the probability of kth class.

Train classifier. The Naive Bayes classifier will encounter view insufficiency. The view is insufficient when there exist instances $(\boldsymbol{x_i}, y_i^t)$ or $(\boldsymbol{y}_i, y_i^t)$, on which the posterior probability $P(y_i^t = k|\boldsymbol{x_i})$ or $P(y_i^t = k|\boldsymbol{y}_i)$ is not equal to 1 or 0 due to the insufficient information provided by $\boldsymbol{x_i}$ or $\boldsymbol{y}_i$ for predicting the label. For these instances, the optimal classifier can not perfectly predict their labels because of the view insufficiency. This will raise the noises of pseudo labels from begin to end (Wang and Zhou 2013). In our approach, the learning view from labels is a highly insufficient view, and for every instance $P(y_i^t = k|\boldsymbol{y}_i)$ is not equal to 1 or 0, if each label is noisy. Besides, the learning view from data is also an insufficient view in some cases, due to feature corruption or feature noises. To minimize the empirical risk, the algorithm should search the classifier which has the lowest observed inconsistent training instances, so the key is to prevent the propagation of incorrect labels.

We propose small-loss metric to measure label confidence for reliable instance selection. That is to say, each learning view chooses small-loss instances from another view's pseudo labels for itself to train in every round. Its effectiveness has been demonstrated in (Meng, Zhao, and Jiang 2015; Ma et al. 2017; Han et al. 2018; Jiang et al. 2018), as it can select more clean instances effectively.

In the learning view from labels, as Naive Bayes usually get parameters by maximum likelihood estimation (MLE), we can get its loss function with small-loss metric as:

$$\ell_l\left(\mathbf{y}, \boldsymbol{y}^d; \boldsymbol{\pi}, \boldsymbol{q}\right) = \sum_{i \in \boldsymbol{I}^l(\alpha_l)} \ell\left(\phi_l(\boldsymbol{y}_i; \boldsymbol{\pi}, \boldsymbol{q}), \mathbf{h}(y_i^d)\right), \tag{3}$$

where $\ell(.)$ is the cross-entropy loss function; $\mathbf{h}(.)$ is the one-hot function; y_i^d is the predicted label of the ith instance; $\boldsymbol{I}^l$ is ϕ_l's confident small-loss instance set and that is to say, $\boldsymbol{I}^l = \arg\min_{\boldsymbol{I}:|\boldsymbol{I}|>\alpha_l|\mathbf{y}|} \sum_{i \in \boldsymbol{I}} \ell(\phi_l(\boldsymbol{y}_i), \mathbf{h}(y_i^d))$ and the setting of ratio coefficient $\alpha_l \in (0, 1]$ will be discussed later. Then $\boldsymbol{\pi}$ is acquired by Eq. (4). Note that, $\boldsymbol{I}^l$ is updated every epoch.

$$\pi_{ks}^{(j)} = \frac{\sum_{i \in \boldsymbol{I}^l(\alpha_l)} \sum_{j=1}^{m} \mathbb{I}\left[y_i^d = k\right] \left[y_{ij} = s\right]}{\sum_{i \in \boldsymbol{I}^l(\alpha_l)} \mathbb{I}\left[y_i^d = k\right]}. \tag{4}$$

Besides, we set $q_k = 1/c$, in order to balance class.

Update aggregated labels. In every round, the view updates its own pseudo labels $\boldsymbol{y}^l$ as:

$$y_i^l = \arg\max_j \phi_l(\boldsymbol{y}_i^l)_j, i = 1, 2, ..., n. \tag{5}$$

3.3 Learning View from Data

Base model. As deep network has a high capacity to learn from data, we regard deep networks as data classifier.

Train classifier. As well as small-loss metric, we further propose two strategies in this view. First, due to high capacity of deep networks in fitting all data (Zhang et al. 2016), it may introduce high accumulation of initialization-induced error biases even after the usage of small-loss metric. Inspired by (Han et al. 2018), we further employ co-teaching strategy. It means that ϕ_d uses two networks ϕ_{d1} and ϕ_{d2} with same structure but different initialization, and in every "batch" each network regards its selected small-loss instances as useful knowledge, and teaches such instances to its peer network for updating the parameters. Since two networks have different learning abilities, they can filter out different types of errors introduced by noisy labels. In this exchange procedure, the error can be reduced by peer networks mutually. Second, as the selected instances is usually class-imbalanced, a class balance constraint is necessary. For deep classifier, since our method selects instances in every mini-batch, we make a dynamic class-weighted loss in every mini-batch to prevent network from excessive preference for some particular classes.

During training, we first shuffle and divide $\{\mathbf{x}, \boldsymbol{y}^l\}$ into p mini-batch $\{\mathbf{x}_k, \boldsymbol{y}_k^l\}$, $k = 1, 2, ..., p$ in every epoch. Then in kth batch $\{\mathbf{x}_k, \boldsymbol{y}_k^l\}$, we train ϕ_{d1} and ϕ_{d2} on each other's confident small-loss instance set $\boldsymbol{I}_k^{d2}$ and $\boldsymbol{I}_k^{d1}$ with a class-weighted loss, respectively. The loss function in kth batch can be written as Eq. (6).

$$\begin{aligned} &\ell_{dk}\left(\mathbf{x}_k, \boldsymbol{y}_k^l; \mathbf{w}_{d1}, \mathbf{w}_{d2}\right) = \\ &\sum_{i \in \boldsymbol{I}_k^{d2}(\alpha_d)} w_{y_i^l}^{d2} \ell(\phi_{d1}(\boldsymbol{x}_i; \mathbf{w}_{d1}), \mathbf{h}(y_i^l)) + \lambda_d ||\mathbf{w}_{d1}||_2^2 \\ + &\sum_{j \in \boldsymbol{I}_k^{d1}(\alpha_d)} w_{y_j^l}^{d1} \ell(\phi_{d2}(\boldsymbol{x}_j; \mathbf{w}_{d2}), \mathbf{h}(y_j^l)) + \lambda_d ||\mathbf{w}_{d2}||_2^2, \end{aligned} \tag{6}$$

where $\mathbf{w}_{d1}$ and $\mathbf{w}_{d2}$ are the learnable parameters of ϕ_{d1} and ϕ_{d2}, respectively; $\boldsymbol{I}_k^{d1} = \arg\min_{\boldsymbol{I}:|\boldsymbol{I}|>\alpha_d|\mathbf{x}_k|} \sum_{i \in \boldsymbol{I}} \ell(\phi_{d1}(\boldsymbol{x}_i), \mathbf{h}(y_i^l))$, $\boldsymbol{x}_i \in \mathbf{x}_k$ and the setting of ratio coefficient $\alpha_d \in (0, 1]$ is discussed later; w_y^{d1} is the weight of yth class in $\boldsymbol{I}_k^{d1}$, and it can be simply defined as $w_y^{d1} = \frac{r_y}{\sum_{j=1}^{c} r_j}$, where $r_j = \frac{1}{n_j}$ and n_j is the number of jth class instances in $\boldsymbol{I}_k^{d1}$; $\boldsymbol{I}_k^{d2}$ and w_y^{d2} are similar with $\boldsymbol{I}_k^{d1}$ and w_y^{d1}, except that they are from the viewpoint of ϕ_{d2}; Note that w_y^1 and w_y^2 change every batch, and $\boldsymbol{I}_k^{d1}$ and $\boldsymbol{I}_k^{d2}$ are updated every epoch. And the whole loss function $\ell_d\left(\mathbf{x}, \boldsymbol{y}^l; \mathbf{w}_{d1}, \mathbf{w}_{d2}\right)$ is equal to $\sum_{k=1}^{p} \ell_{dk}\left(\mathbf{x}_k, \boldsymbol{y}_k^l; \mathbf{w}_{d1}, \mathbf{w}_{d2}\right)$.

Update predicted labels. In every round, when $y_i^{d1} = \arg\max_j \phi_{d1}(\boldsymbol{x}_i)_j$ and $y_i^{d2} = \arg\max_j \phi_{d2}(\boldsymbol{x}_i)_j$, the learning view updates its own pseudo labels $\boldsymbol{y}^d$ by Eq. (7).

$$y_i^d = \begin{cases} y_i^{d1}, & \text{if} \quad y_i^{d1} = y_i^{d2}; \\ y_i^d, & \text{else,} \quad i = 1, 2, ..., n. \end{cases} \tag{7}$$

3.4 Algorithm Details

The whole algorithm is summarized in Alg. 1, where the predicted and aggregated labels are first initialized and then the alternate optimization is performed. After the two networks ϕ_{d1} and ϕ_{d2} are trained, one network or their ensemble is used for testing according to the deployment resources.

The setting of ratio coefficients. The ratio coefficients α_d and α_l decide the proportion of selected training instances. Intuitively, they should increase as the pseudo label error rate falls and correct rate rises in each round. Hence, we simply set $\alpha_i = 1 - \beta_i \varepsilon_j$ in each round, where $i = d$ and $j = l$ or $i = l$ and $j = d$. ε_j is the error rate of ϕ_j's pseudo labels and β_i is a hyperparameter. Therefore, our approach needs a way to know the error rate of pseudo labels. In our simulation experiments, we estimate it through a small validation set which has both clean and noisy labels. As for β_i, we also get it through validation set and its optimal value is between 1.0 and 2.0 in our experiments when giving enough instances. If ε_l can't be estimated, we increase α_d each iteration until deep model isn't improved by raising it.

Outlier detection. Besides, as the estimation of ε_j tends to be inaccurate in practice as well as the proportion of correct labels in each batch is not always the same, we add a simple method to detect outliers in selected instances. First, we calculate the mean μ and standard deviation σ of $\phi_{di}(i = 1, 2)$'s losses in $\boldsymbol{I}_k^{di}(\alpha_d)$. Second, we drop out those instances whose loss ℓ' can't satisfy $|\ell' - \mu| < \rho\sigma$, where ρ is an abnormal coefficient. With outlier detection, on mini-batch $\mathbf{x}_k$,$\mathbf{y}_k^l$, the $\boldsymbol{I}_k^{di}(\alpha_d)$ is changed to $\boldsymbol{I}_k^{di}(\alpha_d, \rho)$, which refers to $\boldsymbol{I}_k^{di}(\alpha_d)$ after dropping outliers. And w_y^i is changed to the weight of yth class in $\boldsymbol{I}_k^{di}(\alpha_f, \rho)$.

Algorithm 1 Coupled-view Classifier Learning

Require: training data $\{\mathbf{x}, \mathbf{y}\}$, data classifier $\{\phi_{d1}, \phi_{d2}\}$ and label aggregator ϕ_l; hyper-parameters β_d, β_l, λ_d, ρ; max round t_{max}, max epoch e_{max} and batch size n_p or number of mini-batch p.
Ensure: learned parameters $\{\mathbf{w}_{d1}, \mathbf{w}_{d2}\}$ of $\{\phi_{d1}, \phi_{d2}\}$.

```
Initialize predicted labels y^d and aggregated labels y^l by majority voting from y.
Pretrain {φ_d1, φ_d2} on {x, y^d} and φ_l on {y, y^l}.
for t = 1, 2, ..., t_max do
    Set α_d = 1 − β_d ε_l.
    for e = 1, 2, ..., e_max do
        Shuffle and divide {x, y^d} into p mini-batch.
        for k = 1, 2, ..., p do
            Train {φ_d1, φ_d2} on {x_k, y_k^l} by minimizing Eq. (6) with outlier detection.
        end for
    end for
    if t < t_max then
        Set α_l = 1 − β_l ε_d,
        Train φ_l on {y, y^l} by Eq. (4),
        Update y^d by Eq. (7) and y^l by Eq. (5).
    end if
end for
return {w_d1, w_d2}.
```

4 Experiments

To verify the effectiveness of the proposed approach (denoted as **CVL**), we conduct the experiments on two synthetic datasets (MNIST (LeCun et al. 1998) and CIFAR10 (Krizhevsky 2009)) and one real dataset (LabelMe-AMT (Rodrigues and Pereira 2018)).

4.1 Experiment Setting

Datasets. We use MNIST and CIFAR10 to generate datasets with noisy labels. MNIST is a handwritten digital dataset, which has a training set of 60K instances, and a test set of 10K instances. The CIFAR-10 is an image classification dataset that consists of 60K 32×32 colour images in 10 classes, with 6K images per class. There are 50K training images and 10K test images. Like (Yi and Wu 2019; Han et al. 2018), we retain 10% of the training instances for validation, and corrupt these datasets manually by the noise transition matrix Q, where $Q_{ij} = Pr(y = j|y^t = i)$, given that noisy label y is flipped from clean y^t. In our setting, each instance has serval noisy labels from different Q, to simulate m noisy annotators. We generate weak labels with three types of noise transition matrix, including 1) symmetry flipping which simulates that labelers often correctly annotate but may chooses false labels uniformly at random with probability ε, 2) pair flipping which simulates that labelers may make mistakes only within similar classes with probability ε, and 3) class-wise flipping which simulates that labelers only do good in particular classes but chooses labels uniformly at random for other classes. For each dataset, we further design 3 groups with 3 noisy annotators, resulting in 6 noisy datasets (see Tab. 1).

The real dataset LabelMe-AMT is taken from the 8-class image classification dataset (Rodrigues and Pereira 2018). It consists of a total of 2,688 images, where 1,000 of them are used to obtain labels from Amazon Mechanical Turk. Each image is labeled by an average of 2.547 workers (59 workers in total). 500 images are used for validation. The remaining images are used for testing.

Training setting for synthetic datasets. For ϕ_{d1} and ϕ_{d2}, We adopt a 22-layer VGG-like Convolutional Neural Network (CNN) for training deep classifiers, including three blocks followed by two fully-connected layers with 128 and 10 neurons respectively. Each block consists of three 3×3 convolutional layers following by leaky ReLU activation. The first two blocks contain 128 and 256-channel convolutional layers, and follow max-pooling and 25% rate dropout layers. Average pooling is used after the last block.

For noisy MNIST-i (i=1,2,3) dataset, the learning rate

Table 1: The generated noisy datasets by simulating different types of annotators on MNIST and CIFAR10.

Experiments	1st annotator	2nd annotator	3rd annotator
MNIST-1	Symmetry, $\varepsilon = 0.8$	Symmetry, $\varepsilon = 0.7$	Pair, $\varepsilon = 0.45$
MNIST-2	Symmetry, $\varepsilon = 0.8$	Symmetry, $\varepsilon = 0.7$	Class-wise, correct class 6
MNIST-3	Symmetry, $\varepsilon = 0.6$	Symmetry, $\varepsilon = 0.7$	Class-wise, correct class 8,9
CIFAR10-1	Symmetry, $\varepsilon = 0.8$	Symmetry, $\varepsilon = 0.7$	Pair, $\varepsilon = 0.45$
CIFAR10-2	Symmetry, $\varepsilon = 0.6$	Pair, $\varepsilon = 0.45$	Class-wise, correct class 3,5,7
CIFAR10-3	Symmetry, $\varepsilon = 0.6$	Pair, $\varepsilon = 0.45$	Class-wise, correct class 7,8,9

is 1e-3, $\lambda_d = 0$ and $\beta_d = 1.05$. For noisy CIFAR10-i (i=1,2,3), the learning rate is from 1e-3 down to 7e-6 linearly, $\lambda_d = 1e-4$ and $\beta_d = 1.2$. For all datasets, the batch size is 384, $\beta_l = 1.1$, ρ is from 2 down to 0.5 linearly. We estimate $\varepsilon_j (j = d, l)$ through the validation set which has both clean and noisy labels. We use Adam optimizer to train 200 epochs for all models. The deep classifiers are all initialized pretrained by "co-teaching" (Han et al. 2018) with the initial labels. All the results are reported as the average figures of five trials.

Training setting for LabelMe-AMT. For ϕ_{d1} and ϕ_{d2}, in order to fairly compare with other baselines, we use the pretrained CNN layers of the VGG-16 network and apply only one FC layer (with 128 units and ReLU activations) and one output layer on top with 50% dropout. For ϕ_l, only the noise matrices whose worker provides the label are used for calculation. During training, the learning rate is 1e-4, λ_d is 0 and the total epochs are 50. Besides, the number of pretraining epoch is 20, $\rho = 2$, $\beta_d = 1.8$. As most of the input features in ϕ_l are missing value, we find the optimal $\beta_l = 0$. And we estimate $\varepsilon_j (j = d, l)$ through 10% of training set which has both clean and noisy labels. Note that we just use clean labels to get ε_j, not to train.

4.2 Results

We use the test accuracy of the best during training and the last epoch to evaluate the performance and robustness of all the classifiers. Although we can get two networks to combine a better result, in order to fairly compare with other baselines, we test one network ϕ_{d1} and other methods with the same network structure. We compare our model with several state-of-the-art benchmarks, including (i) **MV** is a common baseline, meaning that the deep networks are trained on datasets with the labels aggregated by majority voting; (ii) **AggNet** (Albarqouni et al. 2016) uses EM algorithm to jointly model workers' skills and data classifier. It treats the real label distribution as a hidden variable and considers the learned model as an estimate of it by using noise confusion matrices; (iii) **Crowd Layer** (Rodrigues and Pereira 2018) adds the crowd layer to the output of common network, and the layer adjusts the gradients coming from the labels of that annotator according to its reliability; and (iv) **MBEM** (Khetan, Anandkumar, and Lipton 2018) is an improved EM algorithm that rewrites the likelihood of EM and considers the estimated true labels as hard labels.

Noisy MNIST. Tab. 2 shows the results on three noisy datasets, where some observations are concluded. First, the common baseline, **MV** that directly learns with the aggregated labels by majority voting, is not effective when the label noises are heavy, even for the best test accuracy during training. This implies that effective label aggregation is very crucial in improving the classifiers. Second, our **CVL** almost outperforms other benchmarks on all three datasets under the evaluation of both test accuracies, showing the effectiveness of our approach. Third, we can find other benchmarks overfit to the noisy labels more or less during later training stage, where their performance in last epoch declines severely, compared with the best results. Due to small-loss metric and co-teaching strategy, the accuracy drop of our **CVL** is very small, showing the robustness of our approach. Fig. 3 shows a clearer result, where our approach gradually converges to the optimal value and does minor oscillation.

Table 2: The test accuracy (%) of the best during training (left) and the last epoch (right) on noisy MNIST datasets. The minimal accuracy improvement is also given.

Approach	MNIST-1	MNIST-2	MNIST-3
MV	88.16/52.10	58.09/29.77	90.29/47.78
AggNet	**99.57**/99.30	97.10/55.57	99.07/81.41
Crowd Layer	99.53/75.01	98.38/41.67	99.14/52.14
MBEM	99.38/98.58	98.12/94.65	99.18/97.45
Our CVL	99.45/**99.33**	**99.02/98.60**	**99.19/98.97**
Min↑	-0.12/ 0.03	0.64 / 3.95	0.01 / 1.52

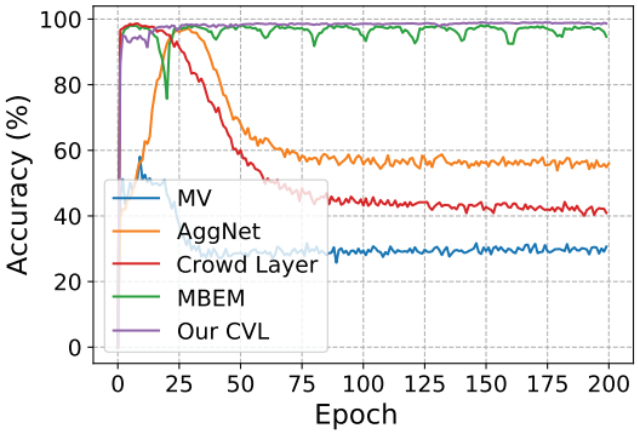

Figure 3: The test accuracy during training on MNIST-2 dataset. It shows the robustness of our approach.

Table 3: The test accuracy (%) of the best during training (left) and the last epoch (right) on noisy CIFAR10 datasets. The minimal accuracy improvement is also given.

Approach	CIFAR10-1	CIFAR10-2	CIFAR10-3
MV	57.58/36.95	67.24/50.49	74.42/53.83
AggNet	83.25/79.56	84.76/83.92	84.45/82.39
Crowd Layer	83.43/58.93	85.52/65.66	76.03/60.59
MBEM	81.55/79.81	85.16/84.05	83.68/82.29
Our CVL	**83.98/83.73**	**86.81/86.67**	**85.42/85.26**
Min↑	0.55 / 3.92	1.29 / 2.62	0.97 / 2.87

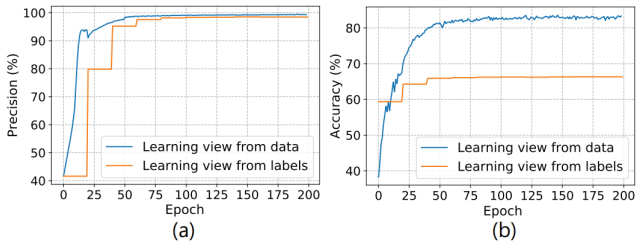

Figure 4: The precision of instance selection (a) and the test accuracy (b) on CIFAR10-1.

Noisy CIFAR10. The performance shows similar results on three noisy CIFAR10 datasets that contain more challenging images, as shown in Tab. 3. From the results, we can find that our **CVL** delivers much better accuracy on all three noisy datasets in terms of both effectiveness and robustness, for example, the minimal improved accuracy reaches 2.62% in last epoch while the maximal accuracy drop is 0.25% between the best and the last accuracy. It empirically demonstrates the mutual learning process in our approach can converge to the good and stable results. Fig. 4 (a) shows the precision of selected pseudo labels in both learning views, while Fig. 4 (b) shows the mutual improvement of two learning views during training. The pure selected labels make model less overfit to false labels, and the selected labels get purer as model improves. This result may partly explain why our approach is better in facilitating good and stable convergence.

Table 4: Test accuracy on LabelMe-AMT.

Approach	Test accuracy (%)
MV	76.744 (± 1.208)
DL-EM (Albarqouni et al. 2016)	82.677 (± 0.981)
MA-sLDAc (Rodrigues et al. 2017)	78.120 (± 0.397)
DL-DN (Guan et al. 2018)	81.888 (± 1.114)
DL-WDN (Guan et al. 2018)	82.410 (± 0.397)
DL-CL (Rodrigues and Pereira 2018)	83.151 (± 0.877)
Our CVL	**86.027 (± 0.313)**

LabelMe-AMT. After the promise is achieved on synthetic datasets, we further check the performance on the real dataset. For fair comparison, we performed 30 executions of our approach for 50 epochs and report average accuracy in the last epoch. Then, we conduct the comparisons with baseline (**MV**) and five state-of-the-art benchmarks, including DL-EM (Albarqouni et al. 2016), MA-sLDAc (Rodrigues et al. 2017), DL-DN (Guan et al. 2018), DL-WDN (Guan et al. 2018) and DL-CL (Rodrigues and Pereira 2018). Tab. 4 lists the results. Our approach achieves the accuracy 86.027 (± 0.313)%, demonstrating its advantages as compared with the other state-of-the-art methods.

Table 5: The test accuracy (%) on CIFAR10-1 under different hyperparameter choices.

β_d/β_l	0.5/1.1	1.0/1.1	1.2/1.1	1.4/1.1	2.0/1.1
Best	76.08	80.15	83.98	**84.03**	73.99
Last	69.94	79.58	83.73	**83.85**	73.76
β_d/β_l	1.2/0.5	1.2/1.0	1.2/1.1	1.2/1.3	1.2/2.0
Best	83.96	84.00	83.98	**84.08**	84.04
Last	83.75	83.78	83.73	**83.80**	83.76

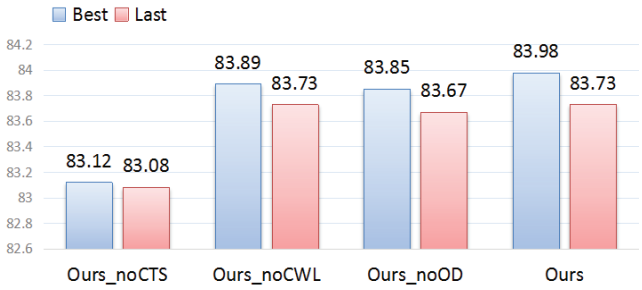

Figure 5: Impact of each component on CIFAR10-1.

4.3 Ablation Study

Choice of hyperparameter β_i. The hyperparameter $\beta_i (i = d, l)$ effects the selection of reliable instances. We test different β_i in α_i on the noisy CIFAR10-1. Tab. 5 lists all the results. When β_i is too small, the selected instances may have a lot of incorrect labels. When β_i is too large, it may drop many difficult but important instances. The result shows that β_d is crucial for good performance, while the result is insensitive to β_l, maybe because neural network has a strong ability to fit all data while Naive Bayes classifier not.

Impact of each component. To study the impact of each component, we use CIFAR10-1 to perform the evaluation. We test our approach without co-teaching strategy ($\text{Ours}_{\text{noCTS}}$), class-weighted loss ($\text{Ours}_{\text{noCWL}}$) and outlier detection ($\text{Ours}_{\text{noOD}}$), respectively. The results are shown in Fig. 5, which shows that the performance of our approach without any component will decline and each component makes positive contributions to the performance.

5 Conclusion

In this paper, we propose a coupled-view learning approach to facilitate deep classifier learning from massive data with multiple noisy labels. It converts a weakly-supervised learning problem into a supervised learning one under mutual supervision that can be solved via alternate optimization. Several strategies are used, which makes our approach robust to label noise and converge stably. Experimental results on

synthetic and real datasets shows the superior performance in effectiveness and robustness. In the future, we will exploit the proposed approach in more tasks.

Acknowledgement This work was partially supported by grants from the National Natural Science Foundation of China (61772513), Beijing Municipal Science and Technology Commission Project (Z191100007119002), National Key Research and Development Plan (2016YFC0801005). Shiming Ge is also supported by Ant Financial through the Ant Financial Science Funds for Security Research, and the Youth Innovation Promotion Association in CAS.

References

Albarqouni, S.; Baur, C.; Achilles, F.; and *et al*. 2016. Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Transactions on Medical Imaging* 35(5):1313–1321.

Amy, M. 2018. Ai researchers embrace bitcoin technology to share medical data. *Nature* 555(7696):293–294.

Bekker, A., and Goldberger, J. 2016. Training deep neural-networks based on unreliable labels. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2682–2686.

Buhrmester, M.; Kwang, T.; and Gosling, S. 2011. Amazonś mechanical turk. *Perspectives on Psychological Science*.

Cheng, J.; Liu, T.; Ramamohanarao, K.; and Tao, D. 2019. Learning with bounded instance-and label-dependent label noise. *arXiv preprint arXiv:1709.03768*.

Cherubin, G. 2019. Majority vote ensembles of conformal predictors. *Machine Learning* 108(3):475–488.

Deng, J.; Guo, J.; Niannan, X.; and *et al*. 2019. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Confernce on Computer Vision and Pattern Recognition*, 4690–4699.

Goldberger, J., and Ben-Reuven, E. 2017. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations*.

Goldman, S., and Zhou, Y. 2000. Enhancing supervised learning with unlabeled data. In *International Conference on Machine Learning*, 327–334.

Guan, M. Y.; Gulshan, V.; Dai, A. M.; and Hinton, G. E. 2018. Who said what: Modeling individual labelers improves classification. In *AAAI Conference on Artificial Intelligence*, 3109–3118.

Han, B.; Yao, Q.; Yu, X.; and *et al*. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, 8536–8546.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *IEEE Confernce on Computer Vision and Pattern Recognition*, 770–778.

Huang, W., and Chen, Y. 2017. The multiset em algorithm. *Statistics & Probability Letters*.

Huang, G. B.; Ramesh, M.; Berg, T.; and Learned-Miller, E. 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Technical report*.

Jiang, L.; Zhou, Z.; Leung, T.; and *et al*. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, 2309–2318.

Khetan, A.; Anandkumar, A.; and Lipton, Z. C. 2018. Learning from noisy singly labeled data. In *International Conference on Learning Representations*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. *Tech Report*.

LeCun, Y.; Bottou, L.; Bengio, Y.; and *et al*. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Li, Y.; Yang, J.; and Song, Y. 2017. Learning from noisy labels with distillation. In *IEEE International Conference on Computer Vision*, 1928–1936.

Liu, T., and Tao, D. 2016. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(3):447–461.

Ma, F.; Meng, D.; Xie, Q.; and *et al*. 2017. Self-paced co-training. In *International Conference on Machine Learning*, volume 70, 2275–2284.

Meng, D.; Zhao, Q.; and Jiang, L. 2015. What objective does self-paced learning indeed optimize? *arXiv preprint arXiv:1511.06049*.

Mitchell, T., and Blum, A. 1998. Combining labeled and unlabeled data with co-training. In *Annual Conference on Computational Learning Theory*, 92–100.

Rodrigues, F., and Pereira, F. C. 2018. Deep learning from crowds. In *AAAI Conference on Artificial Intelligence*, 1611–1618.

Rodrigues, F.; Lourenco, M.; Ribeiro, B.; and Pereira, F. 2017. Learning supervised topic models for classification and regression from crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(12):2409–2422.

Salla, R.; Wilhelmiina, H.; Sari, K.; and *et al*. 2018. Evaluation of the confusion matrix method in the validation of an automated system for measuring feeding behaviour of cattle. *Behav Processes* 148:56–62.

Sun, Z.; Zhang, Q.; Li, Y.; and Tan, Y.-A. 2018. Dppdl: A dynamic partial-parallel data layout for green video surveillance storage. *IEEE Transactions on Circuits and Systems for Video Technology* 28(1):193–205.

Wang, W., and Zhou, Z.-H. 2013. Co-training with insufficient views. In *Asian Conference on Machine Learning*, 467–482.

Wang, W., and Zhou, Z.-H. 2017. Theoretical foundation of co-training and disagreement-based algorithms. *arXiv preprint arXiv:1708.04403*.

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

Yi, K., and Wu, J. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *IEEE Confernce on Computer Vision and Pattern Recognition*, 7017–7025.

Yu, X.; Liu, T.; Gong, M.; and Tao, D. 2018. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision*, 68–83.

Zhang, M.-L., and Zhou, Z.-H. 2011. Cotrade: confident co-training with data editing. *IEEE Transactions on Systems, Man, and Cybernetics* 41(6):1612–1626.

Zhang, C.; Bengio, S.; Hardt, M.; and *et al*. 2016. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.

Zhou, Z.-H., and Li, M. 2005. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering* 17(11):1529–1541.