Safeguarded Dynamic Label Regression for Noisy Supervision

Jiangchao Yao,[†,‡] Hao Wu,[†] Ya Zhang,[†] Ivor W. Tsang,[‡] Jun Sun[†]
[†]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
[‡]Center for Artificial Intelligence, University of Technology Sydney
{sunarker, howiethepeanut, ya_zhang, junsun}@sjtu.edu.cn ivor.tsang@uts.edu.au

Abstract

Learning with noisy labels is imperative in the Big Data era since it reduces expensive labor on accurate annotations. Previous method, learning with noise transition, has enjoyed theoretical guarantees when it is applied to the scenario with the class-conditional noise. However, this approach critically depends on an accurate pre-estimated noise transition, which is usually impractical. Subsequent improvement adapts the pre-estimation in the form of a Softmax layer along with the training progress. However, the parameters in the Softmax layer are highly tweaked for the fragile performance and easily get stuck into undesired local minimums. To overcome this issue, we propose a Latent Class-Conditional Noise model (LCCN) that models the noise transition in a Bayesian form. By projecting the noise transition into a Dirichlet-distributed space, the learning is constrained on a simplex instead of some ad-hoc parametric space. Furthermore, we specially deduce a dynamic label regression method for LCCN to iteratively infer the latent true labels and jointly train the classifier and model the noise. Our approach theoretically safeguards the bounded update of the noise transition, which avoids arbitrarily tuning via a batch of samples. Extensive experiments have been conducted on controllable noise data with CIFAR-10 and CIFAR-100 datasets, and the agnostic noise data with Clothing1M and WebVision17 datasets. Experimental results have demonstrated that the proposed model outperforms several state-of-the-art methods.

Introduction

Large-scale datasets with editorial annotations have greatly driven the success of deep neural networks (DNNs) in computer vision (Krizhevsky, Sutskever, and Hinton 2012), natural language processing (Sutskever, Vinyals, and Le 2014) and speech recognition (Hinton et al. 2012). However, it is usually expensive to collect the clean data in such large volume in many real-world applications. As an alternative, the noisily annotated data on the social websites can be easily acquired inexhaustibly. For example, there are thousands of object photos annotated by social users on the Flickr website. This motivates several works devoted to learning with noisy labels from the deep learning community.

The challenging problem for DNNs in the setting of noisy labels is that it easily memorizes the clean data and the noise simultaneously (Arpit et al. 2017). To overcome this issue, several methods have been explored respectively in the perspective of model regularization, sample re-weighting and noise transition. Arpit et al.(2017) applied the regularization in DNNs to limit its speed of memorizing noise, which prevents the classifier from the noise pollution. Ma et al.(2018) directly corrected the supervision of the classifier by weighting the noisy label with its prediction. However, the methods via model regularization or sample re-weighting usually requires careful hyperparameter selection or network design.

This paper focuses on learning with noise transition. One theoretically grounded work (Patrini et al. 2017) constructed a pre-estimated noise transition on top of the classifier to reduce the influence of noise. Subsequent improvement (Goldberger and Ben-Reuven 2017) adapts the pre-estimation via a Softmax layer along with the training progress. Although it enjoys theoretical guarantees, the model performance via a Softmax layer depends on highly tweaking and the parameters easily get stuck into undesired local minimums. To overcome this issue, we propose to model the noise transition in a Bayesian form. Specifically, the proposed model, Latent Class-Conditional Noise model (LCCN), embeds the noise transition into a Dirichlet-distributed space to constrain the learning on a simplex instead of some ad-hoc parametric space. Furthermore, a dynamic label regression method is deduced for LCCN to iteratively infer the latent true labels and apply them for the classifier training and the noise modeling. It theoretically safeguards the bounded update of the noise transition to avoid arbitrarily tuning as in (Goldberger and Ben-Reuven 2017) via a batch of samples.

Figure 1 provides the illustration of our safeguarded dynamic label regression for LCCN. As can be seen, images are first input to the classifier to have the prediction of latent true labels. Noisy labels are also forwarded to Bayesian noise modeling to compute the conditional transition of latent true labels. Then, the true labels are sampled based on their product and used to supervise the classifier and refine the noise modeling. The whole model is trained end-to-end and scalable to large datasets. In a nutshell, our main contribution can be summarized into the following three points.

- We propose a Latent Class-Conditional Noise model that embeds the noise transition into a Dirichlet space to avoid non-trivially tweaking, and deduce a dynamic label regression method for the model learning.

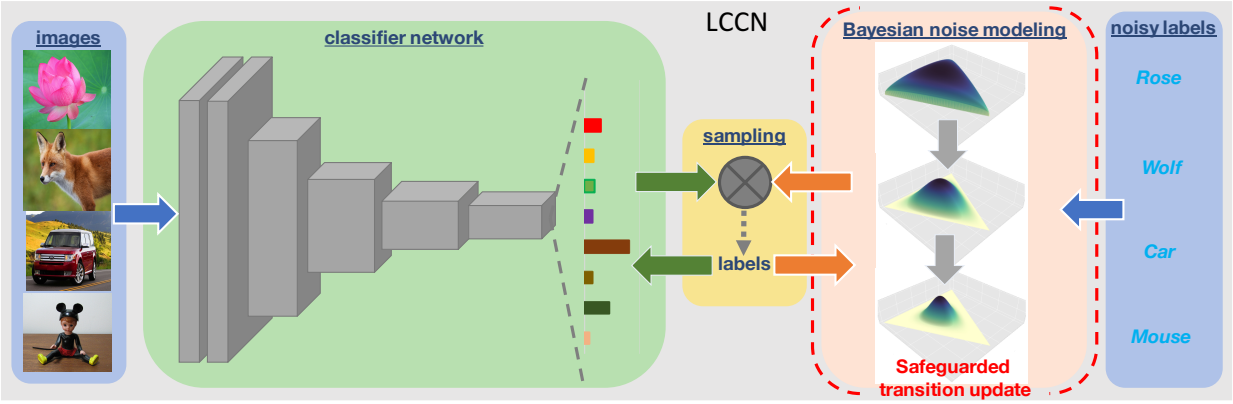

Figure 1: Safeguarded dynamic label regression for LCCN. The images and noisy labels are respectively input to the classifier and the safeguarded Bayesian noise modeling to compute the prediction and the conditional transition. Then, the latent true labels are sampled based on their product and then used for the classifier training and the safeguarded Bayesian noise modeling.

- The classifier is trained based on the sampled posterior labels, which resembles asymptotically training with clean data. Unlike (Goldberger and Ben-Reuven 2017), the optimization of noise transition via a batch of samples is theoretically bounded to avoid arbitrarily tuning.
- We conduct a range of experiments in the small CIFAR-10, CIFAR-100 datasets and the large real-world noisy datasets Clothing1M and WebVision17. The comprehensive results demonstrate the superior performance of our model compared with existing state-of-the-art methods.

Related Work

Recently, several approaches combined with deep learning have been developed for learning with noisy labels. In this section, we simply review these works according to noise transition, sample re-weighting and model regularization.

Learning with Noise Transition This branch of research models a noise transition on top of the classifier to minimize the influence of label noise. Sukhbaatar et al.(2014) first introduced a noise transition matrix on top of CNN to learn with noisy supervision. With a heuristic learning procedure, they gradually make the transition matrix absorb the noise among labels. Misra et al.(2016) investigated the "reporting bias" phenomenon in human-centric annotations by a content-based transition, which is a special case of learning with noisy labels. Patrini et al.(2017) theoretically demonstrated: the backward correction with the inverse of the noise transition is unbiased to train the classifier in the presence of noisy labels; the forward noise transition make the training share the same minimizer with that on the clean data. However, the final performance quite depends on the accuracy of the pre-estimated noise transition. Subsequent improvement in (Goldberger and Ben-Reuven 2017) model the noise transition with a Softmax layer and tune its parameters along with the training progress. Based on this work, Yao et al.(2017) introduced an auxiliary variable to augment the noise transition with more uncertainty. Han et al.(2018a) further added the structure information to constrain the optimization. Although better performance has been achieved, these methods depend on the carefully tweaking. However, our model embeds the noise transition into a Dirichlet space and naturally constrains its optimization to avoid undesired minimums via a dynamic label regression method.

Learning with Sample Re-weighting This line of works weight the contribution of training samples in parameter estimation to reduce the effect of noise (Liu and Tao 2016). It can be implemented by the label or the training pair re-weighting. For example, Reed et al.(2014) facilitated the notion of perceptual consistency to linearly combine the label and the prediction as the new supervision, which shows the substantial robustness to label noise. Subsequently, Li et al.(2017b) substituted the prediction with the refined label by the graph distillation. Wang et al.(2018) leveraged the measure of local intrinsic dimensionality to design an self-weighting strategy for the bootstrapping (Reed et al. 2014). Several works like (Jiang et al. 2018; Ren et al. 2018; Han et al. 2018b) explore to learn a weight or selection for each training pair and then adjust their contribution to the training of the classifier. However, these methods critically depend on the elaborate sample re-weighting strategy.

Learning with Model Regularization This type of methods explore to regularize the training in the presence of noisy supervision. Zhang et al.(2016) have shown that DNNs can easily memorize completely random labels, indicating a serious challenge to learn with noisy labels via DNNs. Their further study (Zhang et al. 2017) that used the convex combinations of images and noisy labels as the data augmentation, has been demonstrated as an efficient regularization to prevent DNNs from overfitting. Arpit et al.(2017) investigated the memorization order of DNNs on feature patterns in noisy datasets and demonstrated dropout can efficiently limit the speed of memorization on noise in DNNs. Tanaka et al.(2018) explicitly introduced a regularization term to prevent the trivial case of assigning all labels to a single class in label correction. Compared with above methods, we indirectly regularize the training by Bayesian noise modeling.

The Proposed Framework

Preliminaries

In the c-class classification setting, a collection of N noisy training pairs $\{(x_n, y_n)\}_{n=1}^N$ is given, where x_n is the raw input data or the feature vector and $y_n \in \{1, \dots, K\}$ is the corresponding noisy label. Assume z_n denotes the true label of x_n, which is unknown in practice. Then the goal in this task is to train a deep network classifier from the noisy dataset $\{(x_n, y_n)\}_{n=1}^N$ analogous to the one trained from the clean dataset $\{(x_n, z_n)\}_{n=1}^N$, so that a good performance can be achieved in a clean test dataset. As shown in (Zhang et al. 2016), directly minimizing the following equation will make DNNs memorize both the classification pattern and noise,

$$f_\theta = \arg\min_{f_\theta \in \mathcal{G}} -\frac{1}{N} \sum_{n=1}^{N} \ell(y_n, f_\theta(x_n)), \tag{1}$$

where f_θ is from the functional space $\mathcal{G}$, which is parameterized by θ via DNNs, and ℓ is the loss function between y_n and the prediction $f_\theta(x_n)$. Equation (1) leads to a bad performance in the clean test dataset since it does not squeeze out the noise influence from f_θ. Therefore, we follows one mainstream of approaches to handle this dilemma, which models a noise transition ϕ in simplex Δ when learning with noisy labels. The objective is then mathematically expressed with the following empirical risk minimization problem

$$f_\theta, \phi = \arg\min_{f_\theta \in \mathcal{G}, \phi \in \Delta} -\frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \phi \circ f_\theta(x_n)), \tag{2}$$

Patrini et al.(2017) theoretically demonstrate Equation (2) trained with the noisy data shares the same minimizer with Equation (1) trained with the clean data, if ϕ is accurately estimated. Unfortunately, it is usually impractical to acquire such a ϕ in advance. Thus, subsequent work (Goldberger and Ben-Reuven 2017) adapts the pre-estimation with a Softmax layer along with the training progress. Although this shows a promising performance, expensive tweaking is usually required due to the brutal-force learning with DNNs.

Latent Class-Conditional Noise model

In this section, we will present our Latent Class-Conditional Noise model (LCCN). Specifically, it avoids non-trivially tweaking for the good performance in (Goldberger and Ben-Reuven 2017) by modeling ϕ in a Bayesian form. The graphical notation is illustrated in Figure 2 and the generative procedure is summarized as follows,

- The latent true label $z_n \sim P(\cdot|x_n)$, where $P(\cdot|x_n)$ is a *Categorical* distribution modeled by the deep neural network f_θ and the given x_n is its input feature.
- The transition vector of the kth class $\phi_k \sim Dirichlet(\alpha)$, where α is the parameter of a $Dirichlet$ distribution and $[\phi_1, \cdots, \phi_K]^T$ constitutes the noise transition matrix.
- The observed noisy label $y_n \sim P(\cdot|\phi_{z_n})$, where $P(\cdot|\phi_{z_n})$ is a *Categorical* distribution parameterized by ϕ_{z_n}.

The general way to solve such a probabilistic model combined with deep learning is amortized variational inference (Kingma and Welling 2014). However, this way for LCCN will require an approximate Categorical reparameterization and introduce an unstable Digamma function to optimize. To avoid this, we deduce a dynamic label regression method for optimization and demonstrate its safeguarded update for ϕ.

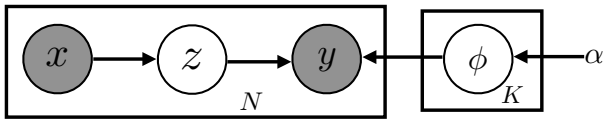

Figure 2: Latent Class-Conditional Noise model. x and y is the observed training pair. z is the latent true label. ϕ is the unknown noise transition. α is a Dirichlet parameter.

Dynamic Label Regression

In the following, we will introduce the dynamic label regression for LCCN, which stacks Gibbs sampling to infer the latent true labels and loss minimization for parameter learning. It naturally suits the case of LCCN and we show its deduction via a two-step formulation. Simply, the first step is computing the probability of each z conditional on the others $Z^{\neg}$[1], i.e., $P(z_n|Z^{\neg n})$. Then with the samples from $P(z_n|Z^{\neg n})$, the classifier training and the noise modeling can be explicitly decoupled as follows,

$$\begin{cases} \min -\frac{1}{n} \sum_{n=1}^{N} \ell_1(z_n, P(z_n|x_n)) \\ \min -\frac{1}{n} \sum_{n=1}^{N} \ell_2(y_n, P(y_n|z_n)). \end{cases} \tag{3}$$

ℓ_1 is the cross-entropy loss and ℓ_2 is the likelihood loss. Alternating between the sampling of $P(z_n|Z^{\neg n})$ and the optimization of Equation (3), we form the algorithm to learn with noisy supervision. Specifically, when $P(z|x)$ approach the true distribution of clean labels, the classifier training is similar to that on the clean dataset. This yields the asymptotically unbiased estimation as on the clean datasets.

Firstly, according to the aforementioned generative process, we can easily deduce the posterior of z conditioned on the observed training pair $\{(x_n, y_n)\}_{n=1}^N$ and the Dirichlet parameter α. This is implemented by factorizing the target conditional probability based on Figure 2 and applying the Bayes theorem as follows,

$$\begin{aligned} & P(Z|X, Y; \alpha) \\ &= \int_\phi \prod_{k=1}^{K} P(\phi_k; \alpha) \prod_{n=1}^{N} P(z_n|x_n, y_n, \phi) d\phi \\ &= \int_\phi \prod_{k=1}^{K} P(\phi_k; \alpha) \prod_{n=1}^{N} \frac{P(z_n|x_n) P(y_n|z_n, \phi)}{P(y_n|x_n)} d\phi \\ &= S * \int_\phi \prod_{k=1}^{K} \frac{\Gamma(\sum_{k'}^{K} \alpha_{k'})}{\prod_{k'}^{K} \Gamma(\alpha_{k'})} \prod_{k'}^{K} \phi_{kk'}^{\alpha_{k'}-1} \prod_{n=1}^{N} \phi_{z_n y_n} d\phi, \end{aligned} \tag{4}$$

where S represents $\prod_{n=1}^{N} \frac{P(z_n|x_n)}{P(y_n|x_n)}$ to simplify above equation. If we use the notation $N_{(\cdot)(\cdot)}$ to represent the confusion

[1]Note that $\neg$ means removing the current object statistic from the whole collection of all object statistics.

matrix of the noisy dataset, then we have $\sum_k^K \sum_{k'}^K N_{kk'}$=$N$ and $\prod_{n=1}^N \phi_{z_n y_n} = \prod_k^K \prod_{k'}^K \phi_{kk'}^{N_{kk'}}$. Putting the later equation into Equation (4) and then using the conjugation characteristic between the Dirichlet distribution and the Multinomial distribution, the following form can be deduced,

$$\begin{aligned} & P(Z|X,Y;\alpha) \\ &= S * \int_\phi \prod_{k=1}^K \frac{\Gamma(\sum_{k'}^K \alpha_{k'})}{\prod_{k'}^K \Gamma(\alpha_{k'})} \prod_{k'}^K \phi_{kk'}^{N_{kk'}+\alpha_{k'}-1} d\phi \\ &= S * \prod_{k=1}^K \frac{\Gamma(\sum_{k'}^K \alpha_{k'})}{\prod_{k'}^K \Gamma(\alpha_{k'})} \prod_{k=1}^K \frac{\prod_{k'}^K \Gamma(\alpha_{k'}+N_{kk'})}{\Gamma(\sum_{k'}^K (\alpha_{k'}+N_{kk'}))}. \end{aligned} \tag{5}$$

Equation (5) is non-analytical and cannot generate the samples of z directly, which motivates the usage of Gibbs sampling. According to the Gibbs sampling, we need to compute $P(z_n|Z^{\neg n})$ first. And then based on $P(z_n|Z^{\neg n})$, a sequence of observations can be sampled, which are approximately from $P(z_n|x_n, y_n, \phi)$. The following deduction facilitates Equation (5) and $\Gamma(x+1) = x\Gamma(x)$ to acquire the final conditional probability for Gibbs sampling.

$$\begin{aligned} P(z_n|Z^{\neg n}, X, Y; \alpha) &= \frac{P(Z|X,Y;\alpha)}{P(Z^{\neg n}|X,Y;\alpha)} \\ &= \frac{P(z_n|x_n)}{P(y_n|x_n)} \frac{\alpha_{y_n} + N_{z_n y_n}^{\neg n}}{\sum_{k'}^K (\alpha_{k'} + N_{z_n k'}^{\neg n})} \\ &\propto \underbrace{P(z_n|x_n)}_{\text{Classifier}} \underbrace{\frac{\alpha_{y_n} + N_{z_n y_n}^{\neg n}}{\sum_{k'}^K (\alpha_{k'} + N_{z_n k'}^{\neg n})}}_{\text{Conditional transition}}. \end{aligned} \tag{6}$$

With Equation (6), we can sample a collection of latent true labels $\{z_n\}$. Such samples are then used to solve the optimization problem in Equation (3). Iterating the procedure of Equation (6) and Equation (3), we gradually approach the true latent label, and at the same time train the classifier and estimate the noise transition. The total algorithm is summarized in Algorithm 1. Note that, we summarize the complete implementation including some details, e.g., pretraining and warming-up, used in the experiments in Algorithm 1.

Safeguarded Transition Update

In this section, we present our analysis to show that our method safeguards the bounded update of the noise transition via a batch of samples, avoiding the arbitrarily tuning with a Softmax layer (Goldberger and Ben-Reuven 2017).

Theorem 1. *Suppose α_i is a positive smoothing scalar, N_i is the current sample number of the ith category (i=1,...,K), M_i is the sum of the sample numbers newly allocated into (positive) and removed from (negative) the ith category after a batch of training samples, and $\widehat{M}_i$ is its absolute sum of such two cases. Then, for the transition vector ϕ_i of the ith category, its variation via a training batch is characterized by the following equation,*

$$\left|\phi_i^{new} - \phi_i^{old}\right| \leq \frac{|r_i| + \widehat{r}_i}{1 + r_i} \tag{7}$$

where $r_i = \frac{M_i}{N_i + \sum_{j=1}^K \alpha_j}$ and $\widehat{r}_i = \frac{\widehat{M}_i}{N_i + \sum_{j=1}^K \alpha_j}$. According to the definition, we have $r_i > -1$, $\widehat{r}_i \geq 0$ and $\widehat{r}_i \geq |r_i|$.

Algorithm 1 Dynamic Label Regression for LCCN

Require: A noisy dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$, a classifier $P(\cdot|x)$ modeled by DNN f_θ, warming-up steps δ, the running epoch number L and the batch-size M.

```
Directly pretrain the classifier f_θ on the noisy dataset D.
Compute the warming-up noise transition matrix φ'.
for epoch i = 1 to L do
    for batch j = 1 to ⌈N/M⌉ do
        Let step=i×⌈N/M⌉+j and hook a batch of samples.
        if step < δ then
            Substitute the transition in Equation (6) with φ', and then sample z_n for each x_n in the batch.
        else
            Sample z_n with Equation (6) for the batch.
        end if
        Update the confusion matrix N_(·)(·) based on the existing sampling observations {(z_n, y_n)}.
        Optimize Equation (3) to learn the classifier f_θ and estimate the noise transition matrix φ.
    end for
end for
Output the classifier f_θ and the noise transition φ.
```

Proof. The variation of ϕ_i after a training batch is,

$$\begin{aligned} & \left|\phi_i^{new} - \phi_i^{old}\right| \\ &= \sum_{j=1}^K \left|\phi_{ij}^{new} - \phi_{ij}^{old}\right| \\ &= \sum_{j=1}^K \left| \frac{N_{ij} + \alpha_j + M_{ij}}{N_i + \sum_{j'=1}^K \alpha_{j'} + M_i} - \frac{N_{ij} + \alpha_j}{N_i + \sum_{j'=1}^K \alpha_{j'}} \right| \\ &\leq \sum_{j=1}^K \frac{\left|(N_i + \sum_{j'=1}^K \alpha_{j'}) M_{ij}\right| + \left|(N_{ij} + \alpha_j) M_i\right|}{(N_i + \sum_{j'=1}^K \alpha_{j'})(N_i + \sum_{j'=1}^K \alpha_{j'} + M_i)} \\ &= \frac{(N_i + \sum_{j'=1}^K \alpha_{j'}) \widehat{M}_i + (N_i + \sum_{j'=1}^K \alpha_{j'}) \left|M_i\right|}{(N_i + \sum_{j'=1}^K \alpha_{j'})(N_i + \sum_{j'=1}^K \alpha_{j'} + M_i)} \\ &= \frac{|r_i| + \widehat{r}_i}{1 + r_i} \end{aligned} \tag{8}$$

□

Corollary 1.1. *Suppose M is the batch size in the training. If it satisfies the condition $M \ll N_i$, we have $\widehat{r}_i < \frac{M}{N_i}$ in a small scale. Then the variation of ϕ_i after a training batch will be bounded by $\frac{|r_i| + \widehat{r}_i}{1 + r_i} \leq \frac{2\widehat{r}_i}{1 - \widehat{r}_i} \approx 2\widehat{r}_i$ in a small scale.*

The core drawback in (Goldberger and Ben-Reuven 2017) is that the noise transition modeled by a Softmax layer can be arbitrarily updated via a batch of samples. It is because the gradients of the parameters in the Softmax layer can be arbitrarily large according to the standard backpropagation. Then, the noise transition might be pushed into a bad local minimum by some extreme noise in a batch of noisy training samples, yielding a serious harm on the classifier train-

ing. The later experimental analysis in Figure 4 can confirm this point. Compared with the way in (Goldberger and Ben-Reuven 2017), our dynamic label regression theoretically safeguards the bounded update of the noise transition via a batch of samples. Specifically, with the bounded update, the conditional transition in Equation (6) is gradually changing towards at a true distribution when the classifier is well trained. Similarly, with more reliable sampled labels, the classifier is better trained and the noise modeling is refined. Finally, we acquire a virtuous cycle for optimization.

Complexity Analysis

Stochastic training a DNN model involves two steps, the forward and backward computations. In each mini-batch update, its time complexity is $\mathcal{O}(M\Lambda)$, where M is the mini-batch size and Λ is the parameter size. Here, in Algorithm 1, we additionally add a sampling operation via Equation (6) whose complexity is $\mathcal{O}(M+K^2)$ (K is the class size). Note that, the first term in the RHS of Equation (6) has been computed in the forward procedure. So the extra cost for the sampling is negligible compared to $\mathcal{O}(M\Lambda)$. An optimization for noise modeling is also negligible, which involves the normalization of a confusion matrix only and the complexity is $\mathcal{O}(K^2)$. Since the big-O complexity of each mini-batch remains the same, our method is scalable to big data.

Experiments

The experiments involve both the simulated noisy datasets and the real-world noisy datasets. We verify the performance of our model by comparing with state-of-the-art methods.

Datasets and Baselines

Datasets We conduct experiments on CIFAR-10, CIFAR-100, Clothing1M and WebVision datasets. CIFAR-10 and CIFAR-100 (Krizhevsky and Hinton 2009) consist of 60,000 32x32 color images respectively from 10 and 100 categories. Both of them contain 50,000 training images and 10,000 test images. We inject the asymmetric noise to disturb their labels to form the noisy datasets. Concretely, on CIFAR-10, we set a probability r to disturb the label to its similar class, i.e., truck $\rightarrow$ automobile, bird $\rightarrow$ airplane, deer $\rightarrow$ horse, cat $\rightarrow$ dog. For CIFAR-100, a similar r is set but the label flip only happens in each super-class. The label is randomly disturbed into the next class circularly within the super-classes. Clothing1M (Xiao et al. 2015) dataset has 1 million images of clothes collected from shopping websites. It has 14 predefined classes and the labels of images are roughly specified based on the surrounding text of images provided by the sellers, thus are very noisy. According to (Xiao et al. 2015), about 61.54% labels are reliable. Besides, this dataset has additional 50k, 14k and 10k clean data respectively for training, validation and test. WebVision[2] (Li et al. 2017a) is a more challenging noisy dataset, which contains more than 2.4 million images crawled from the Internet by using the 1,000 concepts in ILSVRC 2012 as queries. A validation set and a test set, each containing 50,000 annotated images, are also provided to facilitate algorithmic development.

[2]In this paper, we only use the original WebVision 1.0 dataset. The newest version contains more images and more classes.

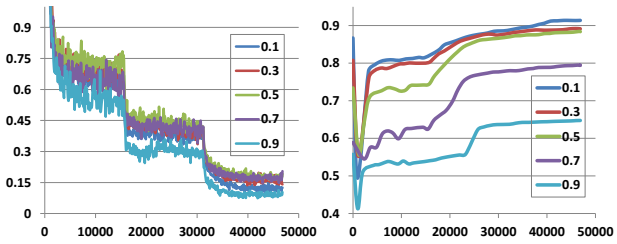

Figure 3: The training loss and the test accuracy of LCCN on CIFAR-10 with the noise level $r = 0.1, 0.3, 0.5, 0.7, 0.9$.

Baselines We compare **LCCN** with the model that directly train the classifier on the dataset (termed as **CE**), the method **Bootstrapping** proposed in (Reed et al. 2014), the transition based method **Forward** (Patrini et al. 2017) and the method that fine-tunes the transition **S-adaptation** (Goldberger and Ben-Reuven 2017). Note that, we choose the hard mode for Bootstrapping, since it is empirically better than the soft mode. For most of the experiments, we apply these four methods as baselines, except that on Clothing1M, we also report the result of **Joint Optimization** (Tanaka et al. 2018), since we adopt the same network configuration and similar learning settings without expensive labor to reproduce.

Implementation

For CIFAR-10 and CIFAR-100, the PreAct ResNet-32 (He et al. 2016) is adopted as the classifier. The image data is augmented by horizontal random flip and 32×32 random crops after padding with 4 pixels. Then the per-image standardization is used to normalize pixel values. For the optimizer, we deploy SGD with a momentum of 0.9 and a weight decay of 0.0005. The batch size is set to 128. The training runs totally 120 epochs and is separated into three phases in 40 and 80 epochs. Among three phases, we respectively use the learning rates 0.5, 0.1 and 0.01. Note that, the reason that we keep the large learning rate (others may set the learning rate smaller than 0.001), is that the small learning rate will lead to overfitting noise as claimed in (Arpit et al. 2017). Following the way in (Patrini et al. 2017), we use CE to initialize the classifier in other baselines and LCCN. For S-adaptation, the following transition is computed to warm-up the transition parameters in the first 80 epochs.

$$\phi'_{ij} = \frac{\Sigma_t 1_{y_t=j} p(z_t = i|x_t)}{\Sigma_t p(z_t = i|x_t)} \tag{9}$$

Similarly on CIFAR-10, we use above transition to warm up the sampling procedure in LCCN for first 20,000 steps. However, on CIFAR-100, we set $\phi'_{ij} = 1[i = j]$ instead since Equation (9) will induce high sampling variance and need long time to converge when there are many classes.

For Clothing1M and WebVision, the ResNet-50 is leveraged as the classifier. We resize the short side of their images to 224 and do the random crop of 224×224. The training

Dataset		CIFAR-10					CIFAR-100				
#	Method \ Noise Ratio	0.1	0.3	0.5	0.7	0.9	0.1	0.2	0.3	0.4	0.5
1	CE	90.10	88.12	76.93	59.01	56.85	66.15	64.31	60.11	51.68	33.37
2	Bootstrapping	90.73	88.12	76.29	57.04	56.79	66.48	64.61	63.01	55.27	**34.52**
3	Forward	90.86	89.03	82.47	67.11	57.29	65.43	62.72	61.28	52.64	33.82
4	S-adaptation	91.02	88.83	86.79	72.74	60.92	65.52	64.11	62.39	52.74	30.07
5	LCCN	**91.35**	**89.33**	**88.41**	**79.48**	**64.82**	**67.83**	**67.63**	**66.86**	**65.52**	33.71
6	CE with the clean data			91.63					69.41		

Table 1: The average accuracy (%) over 5 trials on CIFAR-10 and CIFAR-100 with different noise levels.

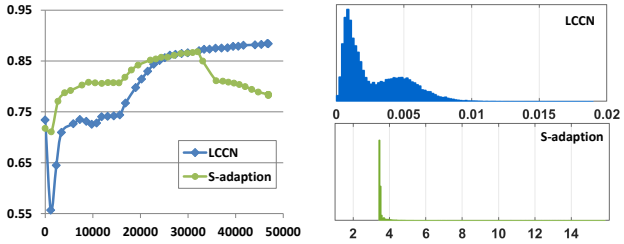

Figure 4: Test accuracy of LCCN and S-adaptation on CIFAR-10 with r=0.5 (left), and the corresponding histogram (right) for the variation of ϕ via a batch of samples.

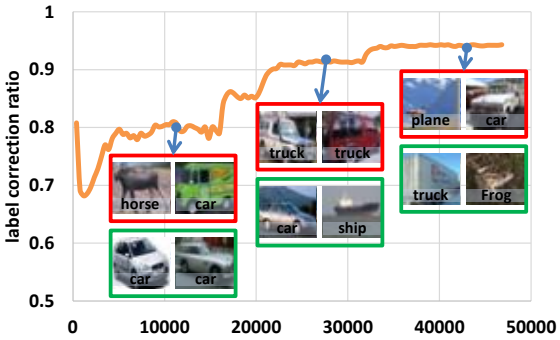

Figure 5: Label correction ratio on CIFAR-10 with r=0.5 and some negative (red box) and positive (green box) correction examples with high probabilities in the training of LCCN.

images are augmented with random flip, whiteness and saturation. For the optimizer, we use SGD with a momentum of 0.9 with a weight decay of 10^{-3}. Same to (Patrini et al. 2017), the batch size for Clothing1M is set to 32 and the corresponding learning rate is initialized with 0.01 for the first 5 epochs and then decreases to 0.001 for the second 5 epochs. Besides, we both validate the performance with warming-up of provided transition (Xiao et al. 2015) and Equation (9). On WebVision, the batch size is set to 128, and the learning rate is initialized with 0.1 is divided by 10 every 30 epochs until 90 epochs. We use the diagonal transition for 10,000 steps of warming-up and then update the confusion matrix to the end, since it contains 1,000 categories.

Results on CIFAR10 and CIFAR-100

Table 1 summarizes the performance of LCCN and baselines on two datasets by averaging their accuracies over 5 trials. From the results, LCCN achieves the best performance at most noise levels. Specifically, even with large r, our model still shows the competitive classification accuracy. For example, when r=0.7 on CIFAR-10 and r=0.4 on CIFAR-100, LCCN reaches 79.48% and 65.52%, outperforming the best results of baselines by about 7% and 13% respectively. This demonstrates that our model is significantly better than baselines. Regarding r=0.5 on CIFAR-100, the way to disturb the labels (Patrini et al. 2017) leads that there is one undesired minimum, since two classes are mixed into one class by equal quota after injecting noise. In this case, it is hard to say which model can achieve the best result. In total, Table 1 shows the superiority of LCCN compared with baselines.

In Figure 3, we trace the training of LCCN on CIFAR-10 with different r to visualize its convergence and robustness. According to Figure 3, we find: *1)* in most cases, i.e., r=0.1, 0.3, 0.5, the relative training loss increases as r increases, while the loss shows attenuation as r>0.5. It is because with the low-level noise, the model can easily correct the labels via the sampling in LCCN. However, when the extreme noise is involved in the dataset, it is more challenging to stop the model from overfitting on noise; *2)* With the training progress, the test accuracy approximately increases and persists to the end of the training. Actually, it is not a common phenomenon for previous methods to own this merit, since all baselines tends to overfitting on noise more or less in the final few epochs. This demonstrates the robustness of LCCN even when training long time with the noisy dataset.

To show LCCN safeguards the noise adaptation compared to S-adaptation, we compute the statistics about their update of noise transition on CIFAR-10 at r=0.5, and illustrate the histogram of changes in Figure 4. Firstly, from the left panel of Figure 4, we can see that there is a significant performance drop in the training of S-adaptation. The clue can be found by inspecting the update of noise transition. As shown in the right panel of Figure 4, the change magnitude of ϕ in S-adaptation is quite high than that of LCCN. This leads to a high risk of over-tuning to undesired local minimums in the presence of noise. Instead, according to the histogram of LCCN, our model updates the noise transition in a small scale, which gradually approaches to a good minimum.

Figure 5 and Figure 6 respectively plot the label correction ratio and the colormap of the confusion matrix when

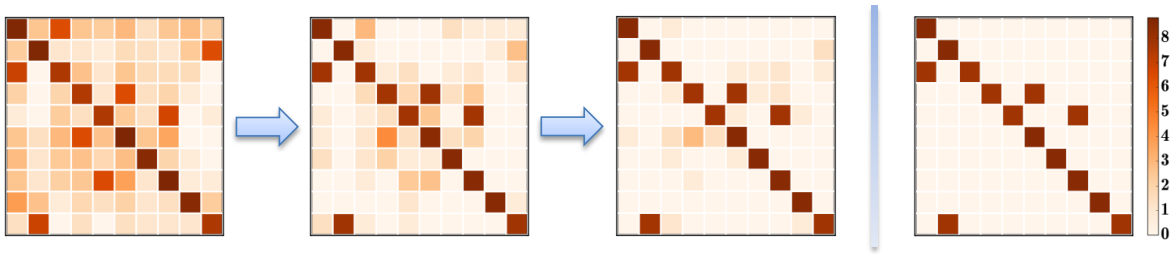

Figure 6: The colormap for the confusion matrix on CIFAR-10 with r=0.5. We use the log-scale of each entry in the confusion matrix for fine-grained visualization. The left three maps are gradually learned by LCCN and the right one is the groudtruth.

#	Method	Accuracy
1	CE	68.94
2	Bootstrapping	69.12
3	Forward	69.84
4	S-adaptation	70.36
5	Joint Optimization	72.23
6	LCCN	**73.07**
7	CE with the clean data	75.28

Table 2: The average accuracy over 5 trials on Clothing1M.

Figure 7: Exemplars between dress and vest on Clothing1M.

training LCCN on CIFAR-10 with r=0.5. As shown in Figure 5, the ratio of the image with the correct label increases along with the training progress. This reflects LCCN successfully models the class-conditional noise and gradually infer the true labels. Specially, by visualizing the miscorrected examples in the training process, we can find that the classifier at first make mistakes in some simple samples, while finally only has the wrong classification in the hard examples. Besides, as can be seen in Figure 6, the initial confusion matrix does not approach the true matrix. However, as the training progresses, the matrix is gradually corrected and at the end of training, it is approximately similar to the ground-truth. These two figures visualize how the safeguarded dynamic label regression method optimizes LCCN.

Results on Clothing1M and WebVision

Table 2 lists the performance of LCCN and baselines on the large-scale Clothing1M. According to the results, we can see that Forward does not show the significant improvement in this dataset, even though they use the manually provided noise transition matrix (Patrini et al. 2017). S-adaptation improves the performance compared with Forward, but only increases by 0.5%. The method that trains the classifier with label correction (Tanaka et al. 2018) has better results than other baselines. However, since they do not model the noise explicitly in their framework, the label correction quite depends on the regularizers to prevent degeneration. Instead, our model that combines the label correction with Bayesian noise modeling, achieves the best performance. Nevertheless, as all baselines facilitate the human auxiliary information, i.e., a manually provided transition matrix or the manually estimated class distribution, it might not be the practical training choice. Thus, we validate our model without the warming-up of the manually estimated transition matrix, where LCCN achieves 71.63%. Furthermore, with a slightly tempering effect of classifier prediction in the sampling, we get 72.92%, which is close to the results with the auxiliary information of LCCN in Table 2. This demonstrates the potential of LCCN in handling the real-world noisy dataset. We present some examples of "dress" and "vest" in Figure 7 to show that images with wrong labels are resigned correctly. As can be seen, the photos of "dress" and "vest" are similar in lack of sleeves, but different in the range to cover the lower body. After training, LCCN successfully learns the difference and infers the corresponding labels for them.

#	Method	Accuracy@1	Accuracy@5
1	CE	63.11	83.69
2	Bootstrapping	63.20	83.81
3	Forward	63.10	83.78
4	S-adaptation	62.54	81.73
5	LCCN	**63.52**	**84.27**

Table 3: The average accuracy over 5 trials on WebVision.

In Table 3, we present the results of LCCN and baselines on a more challenging real-world noisy dataset. Both Top-1 and Top-5 accuracies are reported. According to the results, either in the perspective of Top-1 accuracy or Top-5 accuracy, LCCN achieves the best performance, although the results of all methods do not present the significant gap. One possible reason is in this dataset, there are too many images not belonging to these pre-defined 1000 classes, e.g., images of background noise or crowded scene. Directly imparting these obstacles into the model will seriously disturb the training especially in the presence of so many classes.

Conclusion and Future Work

In this paper, we present a Latent Class-Conditional Noise model to learn with the noisy supervision. Besides, a dynamic label regression method is deployed for LCCN to iteratively infer the latent true labels and jointly train the classifier and model the noise. We theoretically demonstrate that our method safeguards the bounded update of the noise transition to avoid the arbitrarily tuning via a batch of samples. A range of experiments are conducted on both controllable CIFAR-10 and CIFAR-100 datasets and the real-world noisy datasets. Comprehensive results confirm the superior performance of our model compared with existing methods.

Although we have shown the advantages of LCCN in the case of the class-conditional noise, other settings that considers more complex noise should be further explored. This is important since in many applications, the label noise is not only from the known classes, but also from other open set classes. Besides, it is also common that some noise may depend on the content information. To the end, more works based on LCCN can be extended to train with noisy datasets.

Acknowledgements

This work was supported in part by the High Technology Research and Development Program of China (2015AA015801), NSFC (61521062) and STCSM (18DZ2270700), and Australian Research Council grants FT130100746, DP180100106 and LP150100671.

References

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

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

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

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

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

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 29(6):82–97.

Jiang, L.; Zhou, Z.; Leung, T.; Li, L.-J.; and Fei-Fei, L. 2018. Mentornet: Regularizing very deep neural networks on corrupted labels. In *CVPR*.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. *ICLR*.

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

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

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

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

Liu, T., and Tao, D. 2016. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence* 38(3):447–461.

Ma, X.; Wang, Y.; Houle, M. E.; Zhou, S.; Erfani, S. M.; Xia, S.-T.; Wijewickrema, S.; and Bailey, J. 2018. Dimensionality-driven learning with noisy labels. In *ICML*.

Misra, I.; Zitnick, C. L.; Mitchell, M.; and Girshick, R. 2016. Seeing through the Human Reporting Bias: Visual Classifiers from Noisy Human-Centric Labels. In *CVPR*.

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

Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A. 2014. Training deep neural networks on noisy labels with bootstrapping. *ICLR*.

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

Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; and Fergus, R. 2014. Training convolutional networks with noisy labels. *ICLR*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

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

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

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

Yao, J.; Wang, J.; Tsang, I.; Zhang, Y.; Sun, J.; Zhang, C.; and Zhang, R. 2017. Deep learning from noisy image labels with quality embedding. *IEEE Transactions on Image Processing*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2016. Understanding deep learning requires rethinking generalization. *ICLR*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *ICLR*.