Adaptive Sample Selection for Robust Learning under Label Noise

Deep Patel
Advisor: Prof P. S. Sastry

Learning Systems and Multimedia Lab
Department of Electrical Engineering
Indian Institute of Science, Bangalore - 560 012

EECS Symposium
May 2021

Learning under Label Noise

- Given the success of deep learning in the last decade and need for large scale datasets, label errors are inevitable.

*to be defined

Learning under Label Noise

- Given the success of deep learning in the last decade and need for large scale datasets, label errors are inevitable.
- These labelling errors can be due to automated labelling processes, crowdsourced annotations, human errors, etc.

*to be defined

Learning under Label Noise

- Given the success of deep learning in the last decade and need for large scale datasets, label errors are inevitable.
- These labelling errors can be due to automated labelling processes, crowdsourced annotations, human errors, etc.
- The training of deep networks is adversely affected by label noise and hence robust learning* under label noise is an important problem of current interest.

*to be defined

Sample Reweighting – A popular approach

- Many approaches have been proposed to tackle label noise: robust loss functions [8, 50], loss correction [33, 13], meta-learning [41, 23], sample reweighting methods [14, 10, 47, 36], etc.

Sample Reweighting – A popular approach

- Many approaches have been proposed to tackle label noise: robust loss functions [8, 50], loss correction [33, 13], meta-learning [41, 23], sample reweighting methods [14, 10, 47, 36], etc.
- In the last few years, several algorithms proposed for robust learning are based on sample reweighting.

Sample Reweighting – A popular approach

- Many approaches have been proposed to tackle label noise: robust loss functions [8, 50], loss correction [33, 13], meta-learning [41, 23], sample reweighting methods [14, 10, 47, 36], etc.
- In the last few years, several algorithms proposed for robust learning are based on sample reweighting.
- **Idea**: Assign binary or real-valued weights to each sample in the training data and then minimize the weighted loss. When the weights are binary, we call it a sample selection algorithm.

Sample Reweighting – A popular approach

- Many approaches have been proposed to tackle label noise: robust loss functions [8, 50], loss correction [33, 13], meta-learning [41, 23], sample reweighting methods [14, 10, 47, 36], etc.
- In the last few years, several algorithms proposed for robust learning are based on sample reweighting.
- **Idea**: Assign binary or real-valued weights to each sample in the training data and then minimize the weighted loss. When the weights are binary, we call it a sample selection algorithm.
- This is done to reduce the influence of samples that are likely to have noisy labels thereby reducing overfitting to it.

Sample Reweighting – A popular approach

- Many approaches have been proposed to tackle label noise: robust loss functions [8, 50], loss correction [33, 13], meta-learning [41, 23], sample reweighting methods [14, 10, 47, 36], etc.
- In the last few years, several algorithms proposed for robust learning are based on sample reweighting.
- **Idea**: Assign binary or real-valued weights to each sample in the training data and then minimize the weighted loss. When the weights are binary, we call it a sample selection algorithm.
- This is done to reduce the influence of samples that are likely to have noisy labels thereby reducing overfitting to it.
- This idea is very similar to that of 'curriculum learning'.

Sample Reweighting – A popular approach (contd.)

- Curriculum learning – sequencing samples so that 'easy' samples are learned from before 'hard' ones.

Sample Reweighting – A popular approach (contd.)

- Curriculum learning – sequencing samples so that 'easy' samples are learned from before 'hard' ones.
- This notion of 'easy'/'hard' is user-defined.

Sample Reweighting – A popular approach (contd.)

- Curriculum learning – sequencing samples so that 'easy' samples are learned from before 'hard' ones.
- This notion of 'easy'/'hard' is user-defined.
- In the context of label noise, one can think of clean samples as the 'easy' ones and noisy samples as the 'hard' ones.

Sample Reweighting – A popular approach (contd.)

- Curriculum learning – sequencing samples so that 'easy' samples are learned from before 'hard' ones.
- This notion of 'easy'/'hard' is user-defined.
- In the context of label noise, one can think of clean samples as the 'easy' ones and noisy samples as the 'hard' ones.
- This is a plausible analogy as studies such as [3, 9, 27] show that neural networks, when trained on data with label noise, seem to learn from clean data before overfitting to the noisy data.

Sample Reweighting – A popular approach (contd.)

- Curriculum learning – sequencing samples so that 'easy' samples are learned from before 'hard' ones.
- This notion of 'easy'/'hard' is user-defined.
- In the context of label noise, one can think of clean samples as the 'easy' ones and noisy samples as the 'hard' ones.
- This is a plausible analogy as studies such as [3, 9, 27] show that neural networks, when trained on data with label noise, seem to learn from clean data before overfitting to the noisy data.
- Motivated by this, several strategies of 'curriculum learning' have been devised for robustness against label noise

Our Study

- Most methods, in effect, assume that one can assess whether or not an sample has clean label based on some function of the loss value of that sample.

Our Study

- Most methods, in effect, assume that one can assess whether or not an sample has clean label based on some function of the loss value of that sample.
- However, loss value of any specific sample is itself a function of the current state of learning and it evolves with epochs.

Our Study

- Most methods, in effect, assume that one can assess whether or not an sample has clean label based on some function of the loss value of that sample.
- However, loss value of any specific sample is itself a function of the current state of learning and it evolves with epochs.
- Loss values of even clean samples may change over a significant range during the course of learning for different classes.

Our Study

- Most methods, in effect, assume that one can assess whether or not an sample has clean label based on some function of the loss value of that sample.
- However, loss value of any specific sample is itself a function of the current state of learning and it evolves with epochs.
- Loss values of even clean samples may change over a significant range during the course of learning for different classes.
- In addition, these methods require knowledge of noise rates, access to extra data with clean labels or availability of high computation resources. Quite often, these are not available.

Our Study (contd.)

- Motivated by this, we propose a simple, adaptive curriculum based sample selection strategy called **BA**tch **RE**weighting (**BARE**).

Our Study (contd.)

- Motivated by this, we propose a simple, adaptive curriculum based sample selection strategy called **BA**tch **RE**weighting (**BARE**).
- The idea is to focus on the current state of learning, in a given mini-batch, for identifying the noisy labels in it.

Our Study (contd.)

- Motivated by this, we propose a simple, adaptive curriculum based sample selection strategy called **BA**tch **RE**weighting (**BARE**).
- The idea is to focus on the current state of learning, in a given mini-batch, for identifying the noisy labels in it.
- This is done by using the batch statistics of loss values for a mini-batch to compute the threshold for sample selection.

Our Study (contd.)

- Motivated by this, we propose a simple, adaptive curriculum based sample selection strategy called **BA**tch **RE**weighting (**BARE**).
- The idea is to focus on the current state of learning, in a given mini-batch, for identifying the noisy labels in it.
- This is done by using the batch statistics of loss values for a mini-batch to compute the threshold for sample selection.
- We show through empirical studies the effectivenes of our method.

Robust Learning under Label Noise

- In case of label noise, one has access only to samples from the noisy distribution, $\mathcal{D}_\eta$, and not the clean distribution, $\mathcal{D}$.

Robust Learning under Label Noise

- In case of label noise, one has access only to samples from the noisy distribution, $\mathcal{D}_\eta$, and not the clean distribution, $\mathcal{D}$.
- So, for robust learning under label noise, the objective is to learn a classifier from noisy training data, S_η, drawn from $\mathcal{D}_\eta$, such that the classifier performs well on clean training data, S, drawn from $\mathcal{D}$. Here,

Robust Learning under Label Noise

- In case of label noise, one has access only to samples from the noisy distribution, $\mathcal{D}_\eta$, and not the clean distribution, $\mathcal{D}$.
- So, for robust learning under label noise, the objective is to learn a classifier from noisy training data, S_η, drawn from $\mathcal{D}_\eta$, such that the classifier performs well on clean training data, S, drawn from $\mathcal{D}$. Here,
  - $S = \{(\boldsymbol{x}_i, y_i^{cl})\}_{i=1}^m$ – training set with true labels
  - $S_\eta = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ – training set with noisy labels

Modelling Label Noise

The y_i here are the corrupted or noisy labels and they are random variables dependent on the clean labels, y_i^{cl}, through the following conditional probability relations:

$$\eta_{kk'} = P(y_i = e_{k'} | y_i^{cl} = e_k) \tag{1}$$

- These conditional probabilities are called noise rates.

Modelling Label Noise

The y_i here are the corrupted or noisy labels and they are random variables dependent on the clean labels, y_i^{cl}, through the following conditional probability relations:

$$\eta_{kk'} = P(y_i = e_{k'} | y_i^{cl} = e_k) \tag{1}$$

- These conditional probabilities are called noise rates.
- This noise model is referred to as **class-conditional noise**.

Modelling Label Noise

The y_i here are the corrupted or noisy labels and they are random variables dependent on the clean labels, y_i^{cl}, through the following conditional probability relations:

$$\eta_{kk'} = P(y_i = e_{k'} | y_i^{cl} = e_k) \tag{1}$$

- These conditional probabilities are called noise rates.
- This noise model is referred to as **class-conditional noise**.
- A special case of this is **symmetric noise** where we assume $\eta_{kk} = (1 - \eta)$ and $\eta_{kk'} = \frac{\eta}{K-1}$ $\forall k' \neq k$. η here represents the probability of a 'wrong' label.

Modelling Label Noise

The y_i here are the corrupted or noisy labels and they are random variables dependent on the clean labels, y_i^{cl}, through the following conditional probability relations:

$$\eta_{kk'} = P(y_i = e_{k'} | y_i^{cl} = e_k) \tag{1}$$

- These conditional probabilities are called noise rates.
- This noise model is referred to as **class-conditional noise**.
- A special case of this is **symmetric noise** where we assume $\eta_{kk} = (1 - \eta)$ and $\eta_{kk'} = \frac{\eta}{K-1}$ $\forall k' \neq k$. η here represents the probability of a 'wrong' label.
- Assumption: $\eta_{kk} > \eta_{kk'}$, $\forall k' \neq k$

An adaptive curriculum

- General curriculum can be viewed as minimization of a weighted loss [20, 14]:

$$\min_{\theta, \mathbf{w} \in [0,1]^m} \mathcal{L}_{\text{wtd}}(\theta, \mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} w_i \mathcal{L}(f(x_i; \theta), y_i) + G(\mathbf{w}) + \beta ||\theta||^2$$

where $G(\mathbf{w})$ represents the curriculum, $f(\cdot; \theta) \in \Delta^{K-1}$ ($\Delta^{K-1} \subset [0,1]^K$ is the probability simplex) is a classifier function parameterized by θ, and $\mathcal{L}(f(\cdot; \theta), \cdot)$ is a loss function. We use CCE loss here.

An adaptive curriculum (contd.)

- One simple choice for the curriculum is [20] $G(\mathbf{w}) = -\lambda||\mathbf{w}||_1,\ \lambda > 0$. Putting this in the above, omitting the regularization term and taking $l_i = \mathcal{L}(f(x_i;\theta), y_i)$, the optimization problem becomes

$$\begin{aligned}\min_{\theta,\mathbf{w}\in[0,1]^m} \mathcal{L}_{\text{wtd}}(\theta,\mathbf{w}) &= \frac{1}{m}\sum_{i=1}^{m}(w_i l_i - \lambda w_i) \\ &= \sum_{i=1}^{m}(w_i l_i + (1-w_i)\lambda) - m\lambda\end{aligned}$$

An adaptive curriculum (contd.)

- One simple choice for the curriculum is [20] $G(\mathbf{w}) = -\lambda||\mathbf{w}||_1,\ \lambda > 0$. Putting this in the above, omitting the regularization term and taking $l_i = \mathcal{L}(f(x_i;\theta), y_i)$, the optimization problem becomes

$$\begin{aligned}\min_{\theta,\mathbf{w}\in[0,1]^m} \mathcal{L}_{\text{wtd}}(\theta,\mathbf{w}) &= \frac{1}{m}\sum_{i=1}^{m}(w_i l_i - \lambda w_i) \\ &= \sum_{i=1}^{m}(w_i l_i + (1-w_i)\lambda) - m\lambda\end{aligned}$$

- Under the usual assumption that loss function is non-negative, for the above problem, the optimal $\mathbf{w}$ for any fixed θ is: $w_i = 1$ if $l_i < \lambda$ and $w_i = 0$ otherwise.

An adaptive curriculum (contd.)

- One simple choice for the curriculum is [20] $G(\mathbf{w}) = -\lambda||\mathbf{w}||_1,\ \lambda > 0$. Putting this in the above, omitting the regularization term and taking $l_i = \mathcal{L}(f(x_i;\theta), y_i)$, the optimization problem becomes

$$\begin{aligned}\min_{\theta,\mathbf{w}\in[0,1]^m} \mathcal{L}_{\text{wtd}}(\theta,\mathbf{w}) &= \frac{1}{m}\sum_{i=1}^{m}(w_i l_i - \lambda w_i) \\ &= \sum_{i=1}^{m}(w_i l_i + (1-w_i)\lambda) - m\lambda\end{aligned}$$

  - The optimal w_i (for any fixed θ) are still given by the same relation even if λ depend on the class label: for an i with $y_i = e_j$, $w_i = 1$ when $l_i < \lambda_j$. $(\lambda_j = \lambda(e_j))$
  - True even if λ_j a function of θ and of all x_i with $y_i = e_j$ and the current θ.
  - This is what we want. How to do that?

An adaptive curriculum (contd)

- As mentioned earlier, we want these λ_j's to be determined by the statistics of loss values in the mini-batch; equivalently statistics of posterior probabilities.

An adaptive curriculum (contd)

- As mentioned earlier, we want these λ_j's to be determined by the statistics of loss values in the mini-batch; equivalently statistics of posterior probabilities.
- So, we set the sample weights in the following manner:

$$w_i = \begin{cases} 1 & \text{if } f_{y_i}(x_i;\theta) \geq \frac{1}{|\mathcal{S}_{y_i}|}\sum_{s\in\mathcal{S}_{y_i}} f_{y_s}(x_s;\theta) + \sigma_{y_i} \\ 0 & \text{else} \end{cases} \tag{2}$$

where $\mathcal{S}_{y_i} = \{k \in [m] \mid y_k = y_i\}$ and σ_{e_j} indicates the sample variance of the class posterior probabilities for class-j in the given mini-batch.

BARE - The Algorithm

Keeping in mind that the neural networks are trained in a mini-batch manner, the BARE algorithm consists of three parts:

- Computing sample selection threshold, $T_{K\times 1}$, for a given mini-batch of data (Equation 2)
- Sample selection based on this threshold as per Equation 2
- Parameter updation using these selected samples

Experimental Setup

Datasets:

- MNIST [21] (– No data augmentation)
- CIFAR-10 [18] (– random cropping with size 4 padding and random horizontal flips)
- Clothing-1M [43] (– random cropping while ensuring fixed image size)

Table 1: Dataset details

	TRAIN SIZE	TEST SIZE	# CLASS	SIZE
MNIST	60,000	10,000	10	28×28
CIFAR-10	50,000	10,000	10	32×32
CLOTHING-1M	10,00,000	10,000	14	224×224

Experimental Setup (contd.)

Baselines: We compare the proposed algorithm with the following algorithms from literature:

- Co-Teaching (CoT) [10]
- Co-Teaching+ (CoT+) [47]
- Meta-Ren (MR) [36]
- Meta-Net (MN) [37]
- Curriculum Loss (CL) [24]
- Standard (CCE)

Experimental Setup (contd.)

- All the simulations are run for 5 trials.
- All experiments use PyTorch [32], NumPy [12], scikit-learn [34], and NVIDIA Titan X Pascal GPU with CUDA 10.0.

Experimental Setup (contd.)

Types of Label Noise simulated

- Symmetric Label Noise
- Class-conditional Label Noise
  - For MNIST, the following flipping is done: $1 \leftarrow 7$, $2 \rightarrow 7$, $3 \rightarrow 8$, and $5 \leftrightarrow 6$
  - For CIFAR10, the following flipping is done: TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, CAT $\leftrightarrow$ DOG

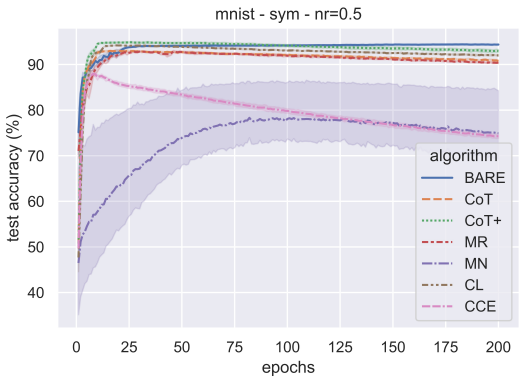
Performance Metrics

- For all algorithms we compare **test accuracies** on a separate test set with clean labels.

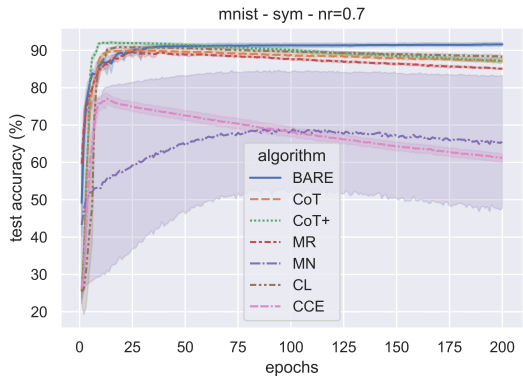
Performance Metrics

- For all algorithms we compare **test accuracies** on a separate test set with clean labels.
- The main idea in all sample selection schemes is to identify noisy labels. Hence, in addition to test accuracies, we also compare:
  - **precision** (# *clean labels selected* / # *of selected labels*)
  - **recall** (# *clean labels selected* / # *of clean labels*)

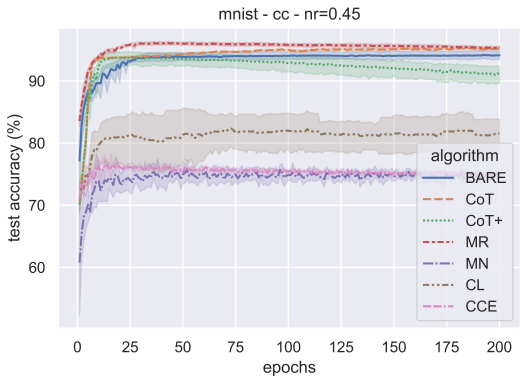
Results on MNIST - Test Accuracy



(a)



(b)

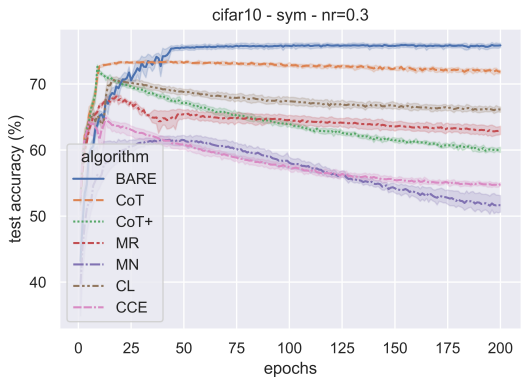


(c)

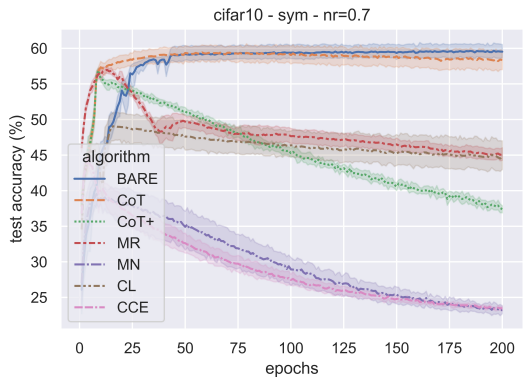
Figure 1: Test Accuracies - MNIST - Symmetric ((a) & (b)) & Class-conditional ((c)) Label Noise

- BARE outperforms the baselines for symmetric noise.
- For class-conditional noise, the test accuracy of BARE is marginally less than the best of the baselines, namely CoT and MR.

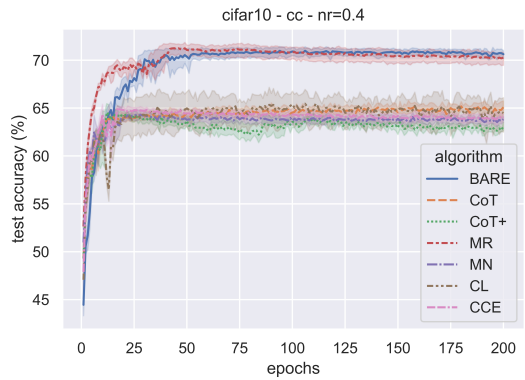
Results on CIFAR10 - Test Accuracy



(a)



(b)



(c)

Figure 2: Test Accuracies - CIFAR10 - Symmetric ((a) & (b)) & ((c)) Class-conditional Label Noise

- BARE outperforms the baseline schemes and its test accuracies are uniformly good for all types of label noise.

Results on Clothing-1M – Test Accuracy

Table 2: Test accuracies on Clothing-1M dataset

Algorithm	Test Accuracy (%)
CCE	68.94
D2L [27]	69.47
GCE [50]	69.75
Forward [33]	69.84
CoT [10]†	70.15
SEAL [6]	70.63
DY [2]	71.00
SCE [40]	71.02
LRT [51]	71.74
PTD-R-V [42]	71.67
Joint Opt. [39]	72.23
BARE (Ours)	**72.28**
DivideMix [22]	74.76

†as reported in [6]

Results on Clothing-1M – Test Accuracy (contd.)

- Even for real-world noisy datasets such as Clothing-1M where the label noise that isn't synthetic unlike that used for simulations on MNIST & CIFAR-10, BARE performs better than all but one baselines.
- Note that DivideMix requires about 2.4 times the computation time required for BARE. And, DivideMix requires tuning of 5 hyperparameters whereas no such tuning is required for BARE.

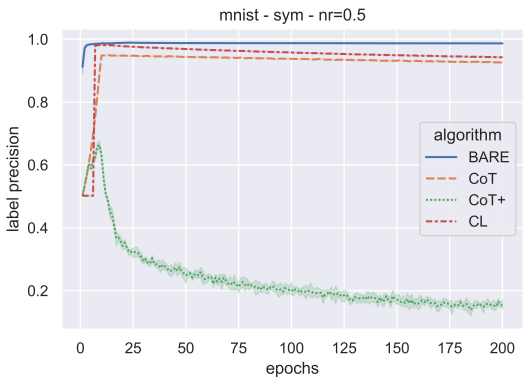
Efficiency of BARE

- Table 3 shows the typical run times for 200 epochs of training with all the algorithms.
- BARE takes roughly the same time as the usual training with CCE loss. Other baselines are significantly more expensive computationally.

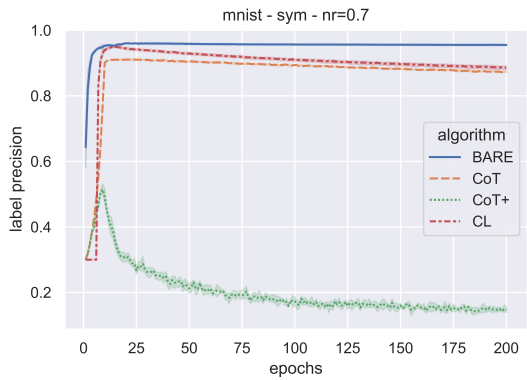
Table 3: Algorithm run times for training (in seconds)

Algorithm	MNIST	CIFAR10
BARE	**310.64**	**930.78**
CoT	504.5	1687.9
CoT+	537.7	1790.57
MR	807.4	8130.87
MN	1138.4	8891.6
CL	730.15	1254.3
CCE	**229.27**	**825.68**

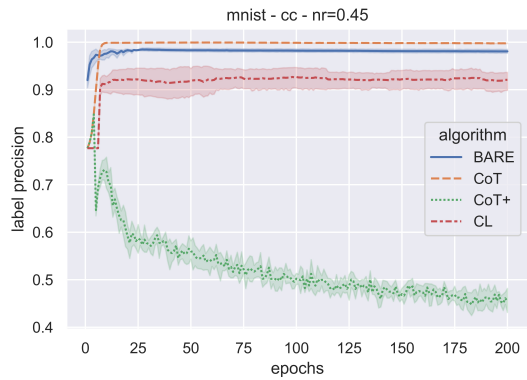
Efficacy of detecting clean samples - Label Precision



(a)



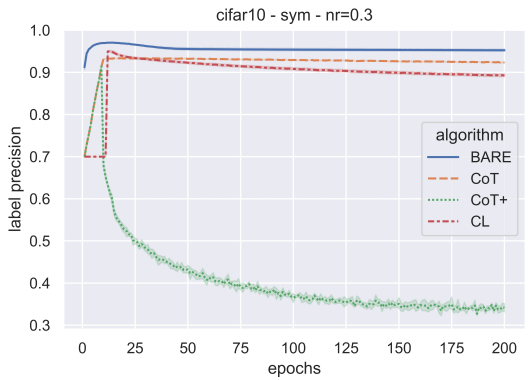
(b)



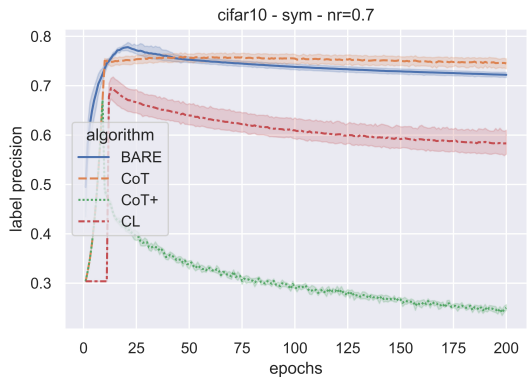
(c)

Figure 3: Label Precision - MNIST - Symmetric ((a) & (b)) & ((c)) Class-conditional Label Noise

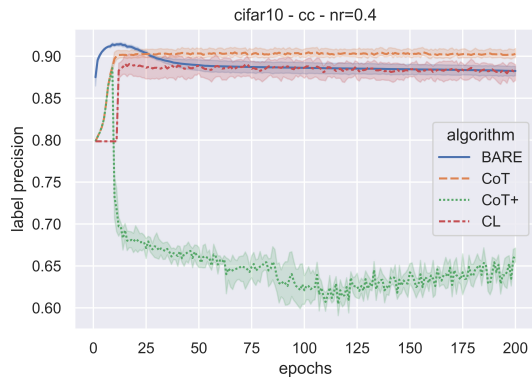
Efficacy of detecting clean samples - Label Precision



(a)



(b)

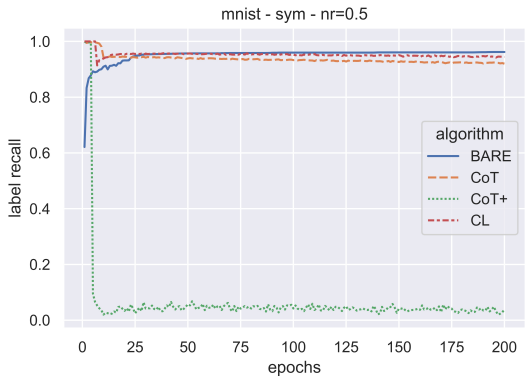


(c)

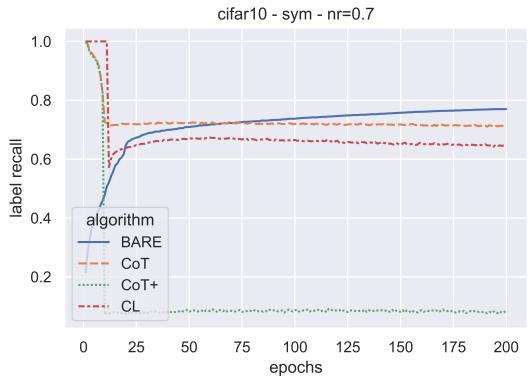
Figure 4: Label Precision - CIFAR10 - Symmetric ((a) & (b)) & ((c)) Class-conditional Label Noise

- Figures 3 and 4 show the label precision (across epochs) on MNIST and CIFAR-10 respectively.
- BARE has comparable or better precision.

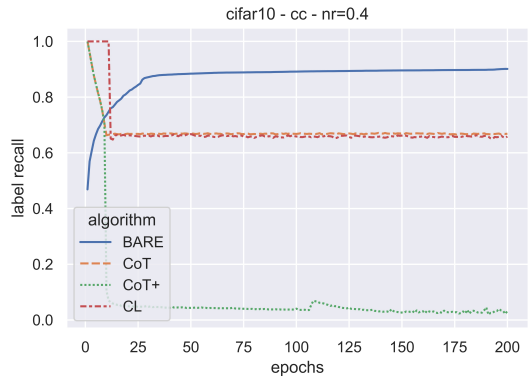
Efficiency of detecting clean samples - Label Recall



(a)



(b)

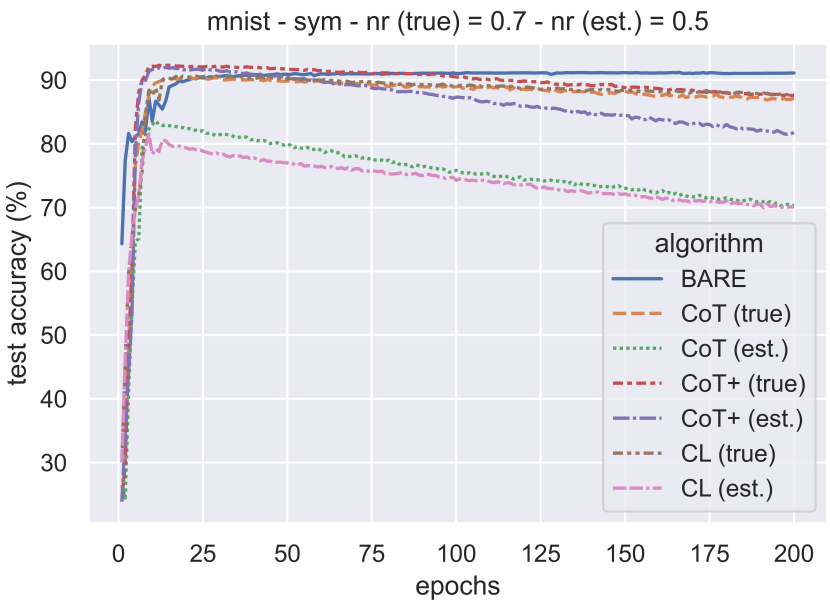


(c)

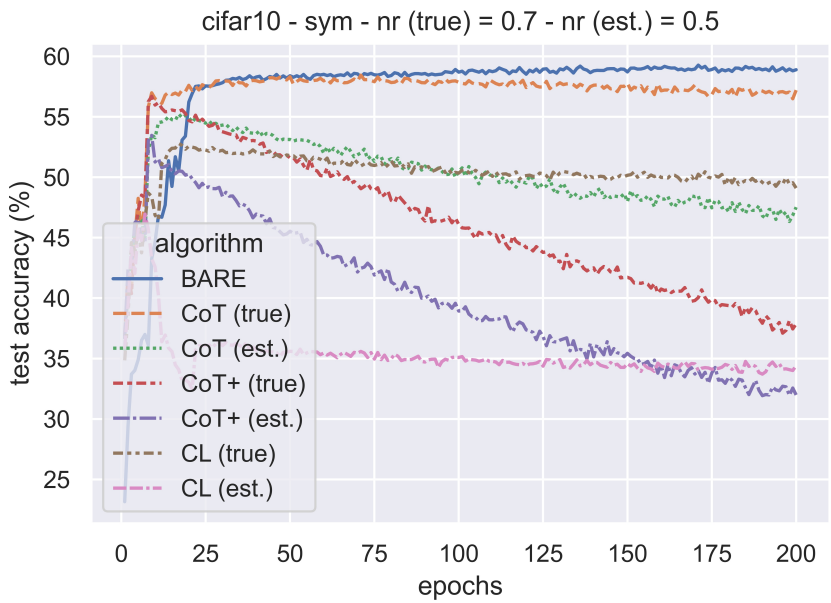
Figure 5: Label Recall - Symmetric ((a) & (b)) & ((c)) Class-conditional Label Noise

- Figure 5 show the label recall values for CoT, CoT+, CL, and BARE for MNIST (5(a)) and CIFAR-10 (5(b) & 5(c)).

Sensitivity to noise rates



(a)



(b)

Figure 6: ((a) & (b)):Test accuracies when estimated (symmetric) noise rate, $\eta = 0.5$, and true noise rate, $\eta = 0.7$, for MNIST & CIFAR-10 resp.

- So, similar performance trends are seen for the case of mis-specified noise rates and arbitrary noise rate matrices.

Conclusions

- We propose an adaptive, data-dependent sample selection scheme, BARE, for robust learning in the presence of label noise.
- The algorithm relies on statistics of assigned posterior probabilities of all samples in a minibatch to select samples from that minibatch.
- The mini-batch statistics are used as proxies for determining current state of learning here.
- Unlike other algorithms in literature, BARE neither needs an extra data set with clean labels nor does it need any knowledge of the noise rates. Further it has no hyperparameters in the selection algorithm.
- Comparisons with baseline schemes on benchmark datasets show the effectiveness of the proposed algorithm both in terms of performance metrics and computational complexity.

Thank You

Any Questions?

References I

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness.
Unsupervised label noise modeling and loss correction.
In *International Conference on Machine Learning*, pages 312–321. PMLR, 2019.

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness.
Unsupervised label noise modeling and loss correction.
In *International Conference on Machine Learning*, pages 312–321. PMLR, 2019.

References II


Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al.
A closer look at memorization in deep networks.
In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.


Carla E Brodley and Mark A Friedl.
Identifying mislabeled training data.
Journal of artificial intelligence research, 11:131–167, 1999.


Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama.
On symmetric losses for learning from corrupted labels.
In *International Conference on Machine Learning*, pages 961–970. PMLR, 2019.

References III

Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng.
Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise.
arXiv preprint arXiv:2012.05458, 2020.

Luis Daza and Edgar Acuna.
An algorithm for detecting noise on supervised classification.
In *Proceedings of WCECS-07, the 1st World Conference on Engineering and Computer Science*, pages 701–706, 2007.

Aritra Ghosh, Himanshu Kumar, and PS Sastry.
Robust loss functions under label noise for deep neural networks.
In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 1919–1925, 2017.

References IV


Jindong Gu and Volker Tresp.
Neural network memorization dissection, 2019.


Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama.
Co-teaching: Robust training of deep neural networks with extremely noisy labels.
In *Advances in neural information processing systems*, pages 8527–8537, 2018.


Sariel Har-Peled, Dan Roth, and Dav Zimak.
Maximum margin coresets for active and noise tolerant learning.
In *IJCAI*, pages 836–841, 2007.

References V

Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant.
Array programming with NumPy.
Nature, 585(7825):357–362, Sept. 2020.

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

References VI


Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei.
Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels.
In *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018.


George H John.
Robust decision trees: Removing outliers from databases.
In *KDD*, volume 95, pages 174–179, 1995.


Amitava Karmaker and Stephen Kwek.
A boosting approach to remove class label noise.
International Journal of Hybrid Intelligent Systems, 3(3):169–177, 2006.


Diederik P Kingma and Jimmy Ba.
Adam: A method for stochastic optimization.
arXiv preprint arXiv:1412.6980, 2014.

References VII


Alex Krizhevsky.
Learning Multiple Layers of Features from Tiny Images.
PhD thesis, University of Toronto, 2009.


H. Kumar and P. S. Sastry.
Robust loss functions for learning multi-class classifiers.
In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 687–692, 2018.


M Pawan Kumar, Benjamin Packer, and Daphne Koller.
Self-paced learning for latent variable models.
In *Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 1*, pages 1189–1197, 2010.


Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner.
Gradient-based learning applied to document recognition.
Proceedings of the IEEE, 86(11):2278–2324, 1998.

References VIII


Junnan Li, Richard Socher, and Steven C.H. Hoi.
Dividemix: Learning with noisy labels as semi-supervised learning.
In *International Conference on Learning Representations*, 2020.


Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli.
Learning to learn from noisy labeled data.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5051–5059, 2019.


Yueming Lyu and Ivor W. Tsang.
Curriculum loss: Robust learning and generalization against label corruption.
In *International Conference on Learning Representations*, 2020.

References IX


Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey.
Normalized loss functions for deep learning with noisy labels.
In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020.


Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey.
Dimensionality-driven learning with noisy labels.
In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018.


Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey.
Dimensionality-driven learning with noisy labels.
In *Proceedings of the 35th International Conference on Machine Learning*, pages 3355–3364, 2018.

References X


Eran Malach and Shai Shalev-Shwartz.
Decoupling" when to update" from" how to update".
In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 961–971, 2017.


Naresh Manwani and PS Sastry.
Noise tolerance under risk minimization.
IEEE transactions on cybernetics, 43(3):1146–1151, 2013.


Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii.
Virtual adversarial training: a regularization method for supervised and semi-supervised learning.
IEEE transactions on pattern analysis and machine intelligence, 41(8):1979–1993, 2018.

References XI

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari.
Learning with noisy labels.
In *Advances in neural information processing systems*, pages 1196–1204, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala.
Pytorch: An imperative style, high-performance deep learning library.
In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

References XII

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu.
Making deep neural networks robust to label noise: A loss correction approach.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay.
Scikit-learn: Machine learning in Python.
Journal of Machine Learning Research, 12:2825–2830, 2011.

References XIII


Scott E Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich.
Training deep neural networks on noisy labels with bootstrapping.
In *ICLR (Workshop)*, 2015.


Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun.
Learning to reweight examples for robust deep learning.
In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4334–4343, 2018.


Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng.
Meta-weight-net: Learning an explicit mapping for sample weighting.
In *Advances in Neural Information Processing Systems*, pages 1919–1930, 2019.

References XIV


Hwanjun Song, Minseok Kim, and Jae-Gil Lee.
Selfie: Refurbishing unclean samples for robust deep learning.
In *ICML*, pages 5907–5915, 2019.


Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa.
Joint optimization framework for learning with noisy labels.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.


Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey.
Symmetric cross entropy for robust learning with noisy labels.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

References XV


Zhen Wang, Guosheng Hu, and Qinghua Hu.
Training noise-robust deep neural networks via meta-learning.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4524–4533, 2020.


Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama.
Part-dependent label noise: Towards instance-dependent label noise.
Advances in Neural Information Processing Systems, 33, 2020.


Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang.
Learning from massive noisy labeled data for image classification.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

References XVI


Linli Xu, Koby Crammer, and Dale Schuurmans.
Robust support vector machine training via convex outlier ablation.
In *AAAI*, volume 6, pages 536–542, 2006.


Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Tin-Yau Kwok.
Searching to exploit memorization effect in learning with noisy labels.
In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10789–10798. PMLR, 2020.


Kun Yi and Jianxin Wu.
Probabilistic end-to-end noise correction for learning with noisy labels.
In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7017–7025, 2019.

References XVII

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama.
How does disagreement help generalization against label corruption?
In *Proceedings of the 36th International Conference on Machine Learning*, pages 7164–7173, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals.
Understanding deep learning requires rethinking generalization.
arXiv preprint arXiv:1611.03530, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz.
mixup: Beyond empirical risk minimization.
arXiv preprint arXiv:1710.09412, 2017.

References XVIII


Zhilu Zhang and Mert R Sabuncu.
Generalized cross entropy loss for training deep neural networks with noisy labels.
arXiv preprint arXiv:1805.07836, 2018.


Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen.
Error-bounded correction of noisy labels.
In *International Conference on Machine Learning*, pages 11447–11457. PMLR, 2020.


Xingquan Zhu, Peng Zhang, Xindong Wu, Dan He, Chengqi Zhang, and Yong Shi.
Cleansing noisy data streams.
In *2008 Eighth IEEE International Conference on Data Mining*, pages 1139–1144. IEEE, 2008.