# Adaptive Sample Selection for Robust Learning under Label Noise


Deep Patel and P S Sastry

Learning Systems & Multimedia Lab, Department of Electrical Engineering, Indian Institute of Science, Bengaluru, INDIA



WACV WAIKOLOA HAWAII JAN 3-7 • 2023

OA$^{\mathrm{M}}$LS 2022


## Learning under Label Noise


- **Given:** Noisily labelled training set, $S_\eta = \{(x_i, y_i)\}_{i=1}^m$, drawn from $\mathcal{D}_\eta$
- ★ **Want:** Learn a classifier to perform well on clean training data, $S = \{(x_i, y_i^{cl})\}_{i=1}^m$, drawn from $\mathcal{D}$
- **Modelling Label Noise:** $y_i$'s (noisy labels) are random variables dependent on the clean labels, $y_i^{cl}$
  - ★ Non-Uniform Noise (NULN):

$$\eta_{\mathbf{x},ij} = \mathbb{P}[y_\mathbf{x} = j | \mathbf{x}, y_\mathbf{x}^{cl} = i] \; \forall i \neq j \in [K], \; \forall \mathbf{x} \quad (1)$$

  - ★ Symmetric Noise (SLN):

$$\eta_{\mathbf{x},ij} = \frac{\eta}{K-1}, \; \eta_{\mathbf{x},ii} = 1 - \eta \; \forall j \neq i, \; \forall \mathbf{x} \quad (2)$$

  - ★ Class-conditional Noise (CCLN):

$$\eta_{\mathbf{x},ij} = \eta_{ij} = \mathbb{P}[y_\mathbf{x} = j | y_\mathbf{x}^{cl} = i] \; \forall j \neq i, \; \forall \mathbf{x} \quad (3)$$

## Sample Reweighting: A popular approach

- Arpit et al. [1] show that neural nets learn from clean data before overfitting to label noise.
- Similarity with *curriculum learning* $\Longrightarrow$ 'cleanly labelled' ≡ 'Easy' & 'noisily labelled' ≡ 'Hard'
- Typically, this 'difficulty' is modelled via sample weights.
- Sample weights = *function*(loss value of that sample)
- ★ **Idea:** Risk Minimization with binary or real-valued sample weights to reduce overfitting

## Proposed Framework: An adaptive curriculum

- ★ **Observation:** Loss value of a sample depends on current state of learning & changes differently per-class throughout training
- ★ **Idea:** Use statistics of samples' loss values in a minibatch to adaptively infer whether a sample has clean labels.

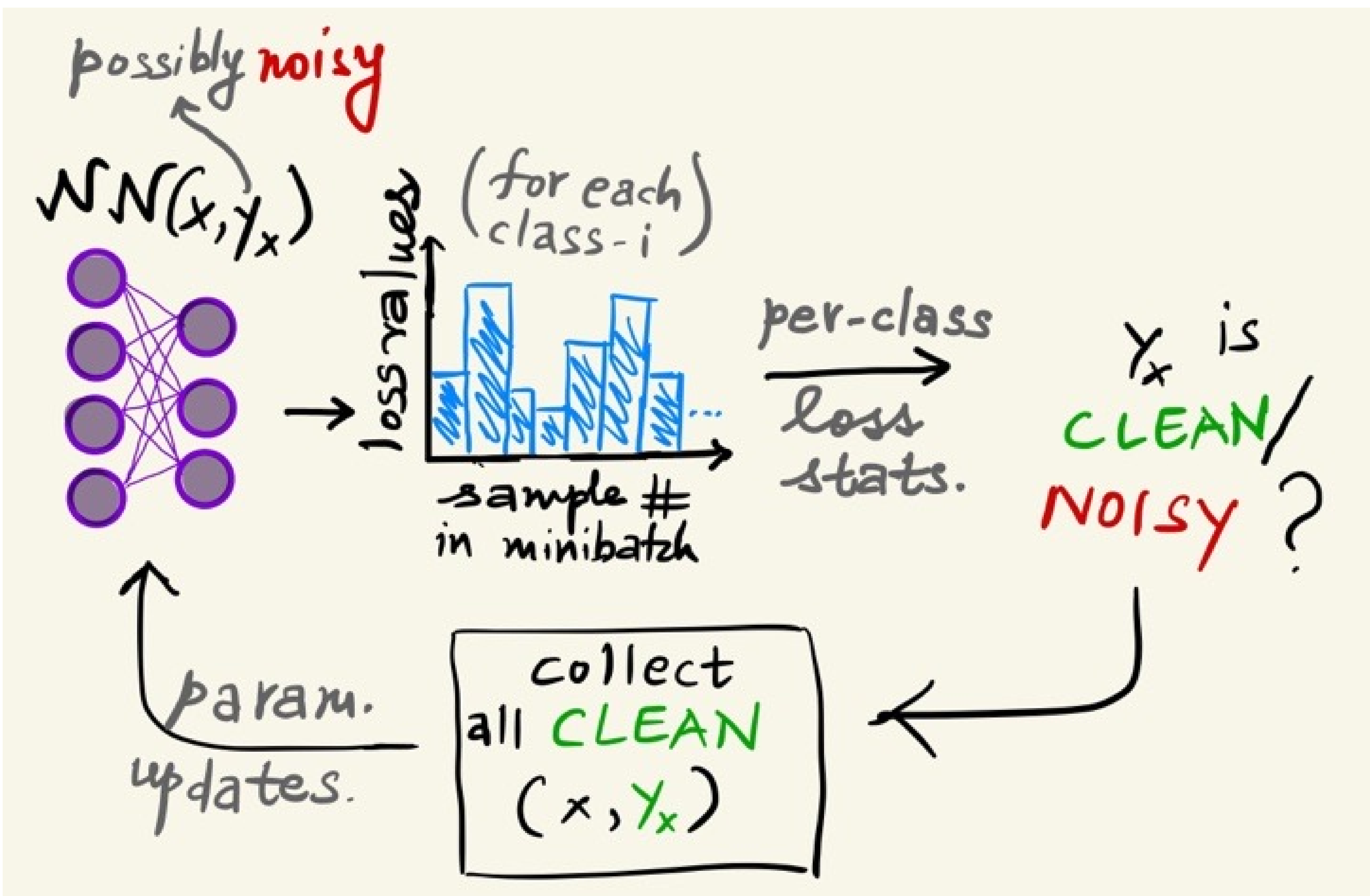


Figure: Outline of the proposed algorithm

## An Adaptive Curriculum

**General Curriculum:**

$$\min_{\theta, \mathbf{w} \in [0,1]^m} \mathcal{L}_{\mathrm{wtd}}(\theta, \mathbf{w}) = \frac{1}{m} \sum_{i=1}^m w_i \mathcal{L}(f(\mathbf{x}_i; \theta), y_i) + G(\mathbf{w}) \quad (4)$$

**Self-Paced Learning [3]:** Use $G(\mathbf{w}) = -\lambda ||\mathbf{w}||_1, \; \lambda > 0$. For a fixed $\theta$, $\forall i \in [m]$, we have:

$$w_i^* = \arg\min_{w_i \in [0,1]} \frac{1}{m} \sum_{i=1}^m w_i \underbrace{\mathcal{L}(f(\mathbf{x}_i; \theta), y_i)}_{\ell_i} - \lambda \cdot w_i = \begin{cases} 1, & \text{if } \ell_i < \lambda \\ 0, & \text{else} \end{cases} \quad (5)$$

**Our Formulation:**

$$w_i^* = \arg\min_{w_i \in [0,1]} \frac{1}{m} \sum_{i=1}^m w_i (\ell_i - \underbrace{\lambda(y_i, \theta, \{\mathbf{x}_j\}_{j=1}^m)}_{\lambda_{y_i}}) = \begin{cases} 1, & \text{if } \ell_i < \lambda_{y_i} \\ 0, & \text{else} \end{cases} \quad (6)$$

We make the thresholds $\lambda_{y_i}$ dependent on current state of learning via:

$$w_i^* = \begin{cases} 1, & \text{if } f_{y_i}(\mathbf{x}_i; \theta) \geq \lambda_{y_i} = \mu_{y_i} + \kappa \cdot \sigma_{y_i} \\ 0, & \text{else} \end{cases} \quad (7)$$

where $\forall p \in [K]$

$$\mu_p := \frac{1}{|\mathcal{S}_p|} \sum_{s \in \mathcal{S}_p} f_p(\mathbf{x}_s; \theta) \text{ and } \sigma_p^2 := \frac{1}{|\mathcal{S}_p|} \sum_{s \in \mathcal{S}_p} (f_p(\mathbf{x}_s; \theta) - \mu_p)^2 \quad (8)$$

**Note:** $\mathcal{S}_p := \{k \in [m] | y_k = e_p\} \; \forall p \in [K]$ where m is the mini-batch size.

**Algorithm:**
**BA**tch **RE**weighting (**BARE**)

```
Input: S_η, # classes K, # epochs T_max, mini-batch size m
Initialize: network parameters, θ_0, for classifier f(·; θ)
while t ≤ T_max do
  Shuffle training set D_η
  for each mini-batch M from D_η do
    for each class p ∈ [K] do
      Compute loss statistics (Equation 8) for S_p = {k ∈ [m] | y_k = e_p}
    end
    R := {(x, y_x) | w*_{y_x} = 1 as per Equation 7}
    θ ← θ − α∇_θ( (1/|R|) Σ_{(x,y_x)∈R} L(x, y_x; θ) )
  end
end
```

## Experimental Setup

- Performance Metrics:
  - ★ Test Accuracy (on a separate test set with clean labels)
  - ★ Label Precision (# clean labels selected / # selected labels)
  - ★ Label Recall (# clean labels selected / # clean labels)
- Datasets:
  - MNIST (SLN/CCLN)
  - CIFAR-10 (SLN/CCLN)
  - Clothing-1M [4] (NULN)
  - Food-101N [2] (NULN)

## Results

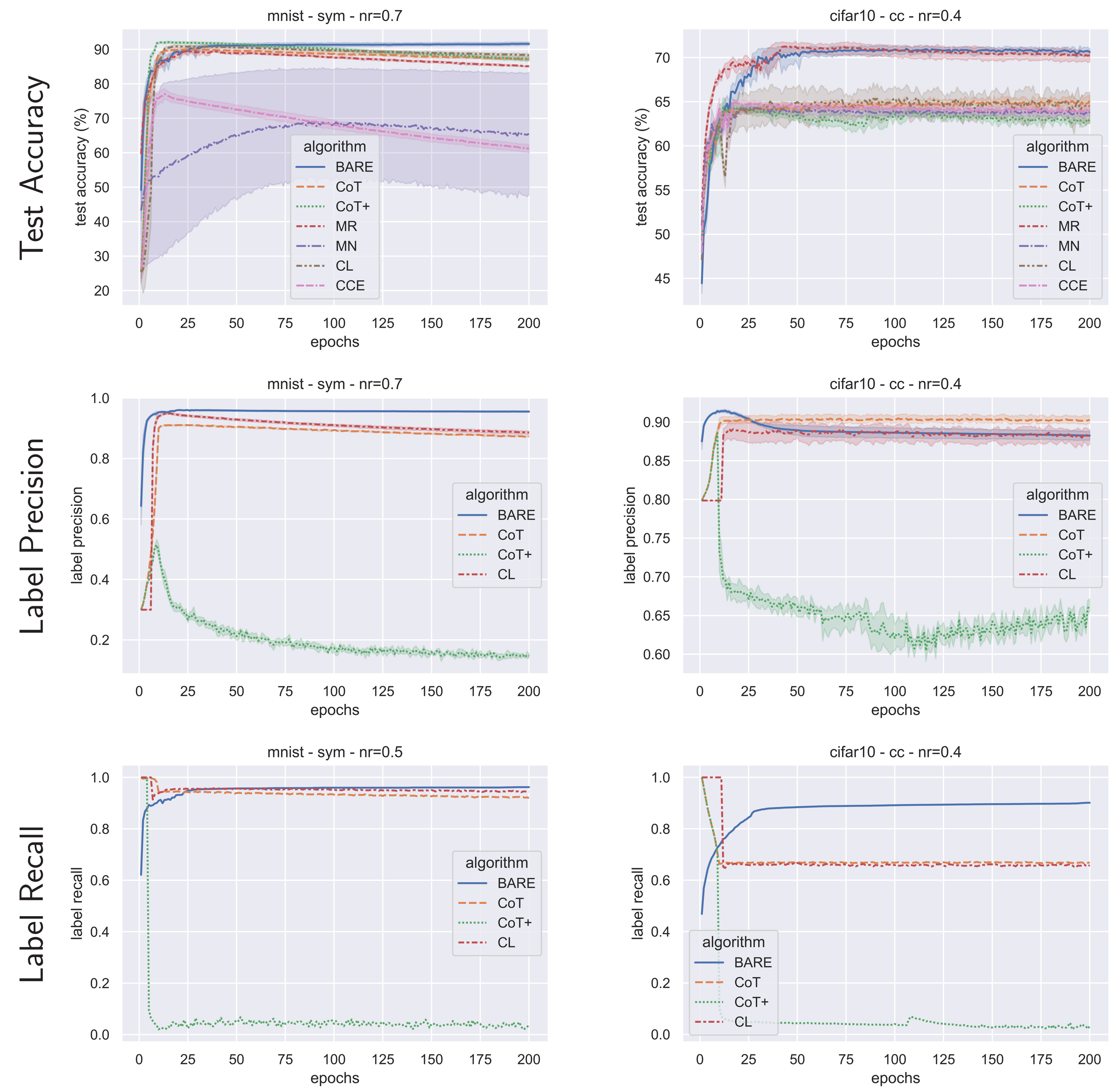


Figure: BARE has comparable or better test accuracies (first row), label precision (second row), and label recall (second row) compared to baselines

## Conclusion

We propose BARE, an adaptive sample selection method, for robust learning under label noise with neural networks.

- Use class-wise statistics of samples' loss values.
- No knowledge of noise rates, extra clean data or training of multiple nets needed.
- On par or improved network performance in terms of test accuracy, label precision, and label recall.

## Acknowledgments


Deep Patel is supported by a Prime Minister's Research Fellowship, Government of India. We thank NVIDIA for providing the Titan X Pascal GPU.


## References


[1] Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. In: International Conference on Machine Learning. pp. 233–242. PMLR (2017)

[2] Bossard, L., Guillaumin, M., Gool, L.V.: Food-101–mining discriminative components with random forests. In: European conference on computer vision. pp. 446–461. Springer (2014)

[3] Kumar, M.P., Packer, B., Koller, D.: Self-paced learning for latent variable models. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 1. pp. 1189–1197 (2010)

[4] Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2691–2699 (2015)



{deeppatel,sastry}@iisc.ac.in


https://github.com/dbp1994/masters_thesis_codes/tree/main/BARE