DRoP: Distributionally Robust Data Pruning

Artem Vysogorets, Kartik Ahuja, Julia Kempe (ICLR 2025)

Presenter: Kyungmin Kim

Efficient ML Systems (EECE695E)

TL;DR

- Data pruning, a process of removing unnecessary data from the original dataset, is known to improve convergence speed, scaling, and resource efficiency.
- Solely focusing on the average performance, authors argue that existing data pruning methods suffer from distributional bias, a performance disparity across different sub-groups of distribution.
- They propose a distributionally-robust data pruning method coined DRoP, which is both theoretically and empirically validated.

Related Work

Related Work: Data Efficiency

- Dataset distillation: Replaces the original samples with synthetically generated counterparts that contain compressed training signal.

- CoreSet method: Selects representative samples that jointly capture the data manifold.

- Data pruning: Removes unnecessary samples in terms of model performance.

Related Work: Data Pruning

- Usually, they design scoring mechanisms to assess the utility of each sample, often measured by its uncertainty or difficulty.

- Data pruning is made in two fold as follows:
  1. Learn a query model ψ, trained on a full training dataset $\mathcal{D} = \{(X_i, y_i)\}_{i \in [N]}$.
  2. Prune the dataset $\mathcal{D}$ based on a utility score $A(X, y; \psi)$ as

$$\mathcal{D}_s := \Big\{(X, y) \in \mathcal{D} : A(X, y; \psi) \geq \text{quantile}\big(\{A(X_i, y_i; \psi)\}_{i \in [N]}, s\big)\Big\}$$

    ✓ Note that a utility score $A(X, y; \psi)$ is defined for each training sample.

Related Work: Data Pruning (cont.)

- Data pruning methods vary by choosing different utility scores.
  - `Forgetting` [1]: The number of times (X, y) is both learned and forgotten while training $\psi(\cdot)$

Algorithm 1 Computing forgetting statistics.

initialize $\text{prev\_acc}_i = 0, i \in \mathcal{D}$
initialize forgetting $T[i] = 0, i \in \mathcal{D}$
while not training done **do**
  $B \sim \mathcal{D}$ # sample a minibatch
  for example $i \in B$ **do**
    compute acc_i
    if $\text{prev\_acc}_i > \text{acc}_i$ **then**
      $T[i] = T[i] + 1$
    $\text{prev\_acc}_i = \text{acc}_i$
  gradient update classifier on B
return T

  - `EL2N` [2]: $A(X, y; \psi) = \|\sigma(\psi(X)) - \mathbf{y}\|_2$, where σ is a softmax function and $\mathbf{y}$ is an one-hot vector.

Related Work: Data Pruning (cont.)

- `Grand` [2]: $A(X, y; \psi) = \|\nabla \mathcal{L}(\sigma_y(\psi(X)), y)\|_2$
- `Dynamic Uncertainty` [3]:
  1. Estimate the variance of the target probability $\{\sigma_y(\psi_j(X))\}_{j=k-J}^{k}$ across a fixed window of J previous epochs, for every training epoch k.
  2. Average across all k.

- Note that a utility score $A(X, y; \psi)$ is defined for each training sample.

Related Work: Distributional Robustness

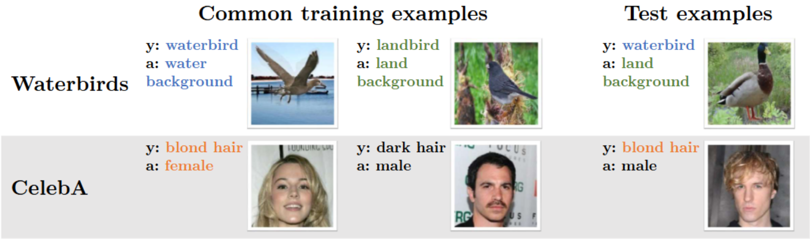
- Distributional robustness in machine learning concerns the distributional bias problem: non-uniform accuracy across different sub-population groups.

- Followings are representative ML problems where distributional robustness matter.

ML Problem	Group Variable
Classification Bias	Class Label
Spurious Correlation	(Spurious Feature, Class Label)
Fairness	(Spurious Feature, Class Label)

  – Certain fairness problems can be considered spurious correlation problems, where the spurious features correspond to demographic attributes.

Related Work: Distributional Robustness (cont.)



- `Waterbirds`: (Water Bg., Water Bird), (Land Bg., Land Bird), (Water Bg., Land Bird), (Land Bg., Water Bird)

- `CelebA`: (Blond Hair, Female), (Black Hair, Male), (Blond Hair, Male), (Black Hair, Female)

Related Work: Distributional Robustness (cont.)

- Many well-established algorithms consider a weighted sum of group-wise expected losses as an objective, aiming to put higher mass on high loss-groups as follows:

$$\min_{\theta \in \Theta} \sum_{g=1}^{G} q_g \underbrace{\mathbb{E}_{(x,y)\sim P_g}\{\ell(\theta;(x,y))\}}_{\text{Expected Loss of Grp. } g}.$$

  - $\theta \in \Theta$: Model Parameter
  - $q := (q_1, ..., q_G)$: Weight vector
  - P_g: Data generating process of group g

- Unlike most group-wise cost weighting strategies that consider a fixed weight vector q [4], Group DRO [5] iteratively updates q for every training step.

Related Work: Distributional Robustness (cont.)

Algorithm 1: Online optimization algorithm for group DRO

Input: Step sizes η_q, η_θ; P_g for each $g \in \mathcal{G}$
Initialize $\theta^{(0)}$ and $q^{(0)}$
for $t = 1, \ldots, T$ **do**
  $g \sim \text{Uniform}(1, \ldots, m)$ `// Choose a group g at random`
  $x, y \sim P_g$ `// Sample x,y from group g`
  $q' \leftarrow q^{(t-1)}$; $q'_g \leftarrow q'_g \exp(\eta_q \ell(\theta^{(t-1)}; (x, y)))$ `// Update weights for group g`
  $q^{(t)} \leftarrow q' / \sum_{g'} q'_{g'}$ `// Renormalize q`
  $\theta^{(t)} \leftarrow \theta^{(t-1)} - \eta_\theta q_g^{(t)} \nabla \ell(\theta^{(t-1)}; (x, y))$ `// Use q to update θ`
end

- Actually, group DRO aims to minimize an expected loss of the worst group, not a weighted sum of group-wise expected losses.

Related Work: Distributional Robustness (cont.)

- In this paper, they mainly consider the classification bias problem.

- Given accuracy r_k for each class $k \in [K]$, the following evaluation metrics are considered:
  - Worst-class accuracy: $\min_k r_k$
  - Difference between the maximum and minimum accuracy: $\max_k r_k - \min_k r_k$
  - Standard deviation: $\mathrm{std}_k\, r_k$

Distributional Bias in Existing Data Pruning Methods

Notation

- `Dataset Density`: The degree of data pruning
- `Class Density`: The degree of data pruning within each class
- `min SPC @10%`: Minimum sample per class at `Dataset Density` 10%
- `Class Accuracy`: Test accuracy for each class evaluated on the model trained will full dataset

Data Pruning is Not Robust

- Authors conducted experiments on class-wise robustness for two computer vision benchmarks, CIFAR-100 and TinyImageNet.

- They considered four different data pruning baselines:
  - `Forgetting` [1]: The number of times (X, y) is both learned and forgotten while training $\psi(\cdot)$
  - `EL2N` [2]: $A(X, y; \psi) = \|\sigma(\psi(X)) - \mathbf{y}\|_2$, where σ is a softmax function and $\mathbf{y}$ is an one-hot vector.
  - `Grand` [2]: $A(X, y; \psi) = \|\nabla \mathcal{L}(\sigma_y(\psi(X)), y)\|_2$
  - `Dynamic Uncertainty` [3]:
    1. Estimate the variance of the target probability $\{\sigma_y(\psi_j(X))\}_{j=k-J}^{k}$ across a fixed window of J previous epochs, for every training epoch k.
    2. Average across all k.

Data Pruning is Not Robust (cont.)

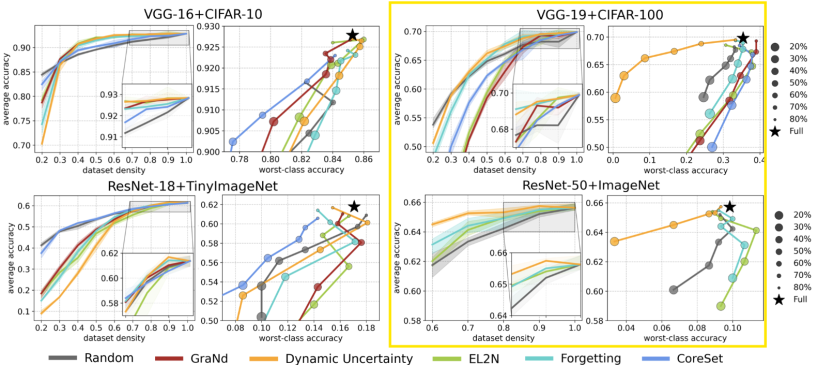


Figure 1: Average test performance of baseline pruning algorithms against dataset density and worst-class accuracy.

Data Pruning is Not Robust (cont.)

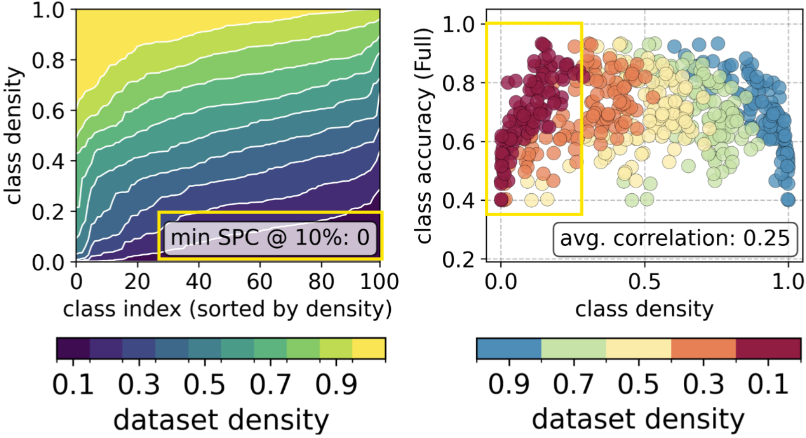


Figure 2: `Dynamic Uncertainty` applied to CIFAR-100. Sorted class densities by dataset density (left). Test class accuracy against class density by dataset density (right).

Theoretical Analysis

Toy Binary Classification Problem

- Authors derived analytical results regarding their proposed method DRoP in a toy binary classification problem.
- Specifically, they consider a linear classification model with a univariate feature $x \in \mathbb{R}$, where a true data generating process is a mixture of two Gaussian distributions as follows:

$$\begin{aligned} p(x) &= \mathbb{P}(y=0) \times p(x|y=0) + \mathbb{P}(y=1) \times p(x|y=1) \\ &= \phi_0 \times \mathcal{N}(\mu_0, \sigma_0^2) + \phi_1 \times \mathcal{N}(\mu_1, \sigma_1^2) \end{aligned}$$

  - Assume $\mu_0 < \mu_1$ and $\sigma_0 < \sigma_1$.

Toy Binary Classification Problem (cont.)

- Let us consider linear decision rules $t \in \mathbb{R} \cup \{\pm\infty\}$ with a prediction function $\hat{y}_t(x) = \mathbb{1}(x > t)$.

- Then, the 0-1 risks of the two classes are as follows:

$$R_0(t) := \mathbb{E}_{x|y=0}\{\hat{y}_t(x) = 1\} = \mathbb{P}_{x|y=0}\{x > t\} = \Phi\left(\frac{\mu_0 - t}{\sigma_0}\right),$$
$$R_1(t) := \mathbb{E}_{x|y=1}\{\hat{y}_t(x) = 0\} = \mathbb{P}_{x|y=1}\{x < t\} = \Phi\left(\frac{t - \mu_1}{\sigma_1}\right),$$

where Φ is a cumulative distribution of the standard normal distribution.

Optimal Decision Rule Minimizing the Average Risk

- Under some technical assumptions on means, variances, and priors, the optimal decision rule minimizing the average risk

$$R(t) = \mathbb{E}_{x,y}\{\hat{y}_t(x) \neq y\} = \phi_0 \times R_0(t) + \phi_1 \times R_1(t)$$

is given as

$$t^*\left(\frac{\phi_0}{\phi_1}\right) = \frac{\mu_0\sigma_1^2 - \mu_1\sigma_0^2 + \sigma_0\sigma_1\sqrt{(\mu_0-\mu_1)^2 - 2(\sigma_0^2-\sigma_1^2)\ \log\frac{\phi_0\sigma_1}{\phi_1\sigma_0}}}{\sigma_1^2 - \sigma_0^2}.$$

- In the balaned case where $\phi_0 = \phi_1 = 0.5$, the heavier-tailed class is more difficult in the sense that

$$R_1\big(t^*(1)\big) > R_0\big(t^*(1)\big).$$

Optimal Decision Rule Minimizing the Average Risk (cont.)

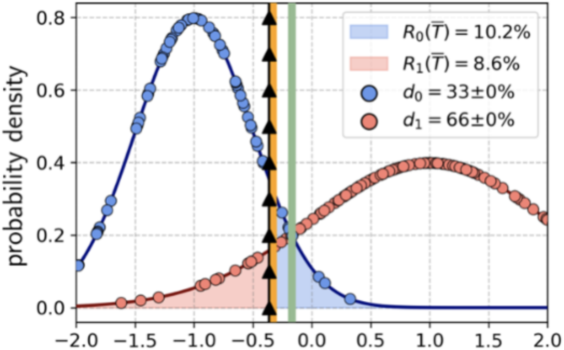


Figure 3: Green line corresponds to the optimal decision rule $t^*\left(\frac{\phi_0}{\phi_1}\right)$ minimizing the average risk $R(t)$.

Optimal Decision Rule Minimizing the Worst-class Risk

- The optimal decision rule minimizing the worst-class risk

$$R_{\text{worst}}(t) = \max\{R_0(t), R_1(t)\}$$

is given as $\hat{t}$ that satisfies $R_0(\hat{t}) = R_1(\hat{t})$.

- Based on the definition of $R_0(t)$ and $R_1(t)$,

$$\hat{t} = (\mu_0\sigma_1 + \mu_1\sigma_0)/(\sigma_0 + \sigma_1).$$

DRoP: Distributionally Robust Data Pruning

- Authors aim to prune the data in a way that average risk minimization achieves the best worst-class risk.

- In other words, they are trying to find a mixture ratio $\frac{\tilde{\phi}_0}{\tilde{\phi}_1}$ that satisfies

$$t^*\left(\frac{\tilde{\phi}_0}{\tilde{\phi}_1}\right) = \hat{t},$$

where $\frac{\sigma_0}{\sigma_1}$ satisfies the condition.

- In terms of optimization, we can adopt ERM objective without concerning much about the classification bias.

DRoP: Distributionally Robust Data Pruning (cont.)

- In practice, letting d_k and N_k be the fraction of samples to be retained and the number of training samples in class k, we aim to find d_0 and d_1 s.t.

$$d_0 N_0 / d_1 N_1 = \sigma_0 / \sigma_1. \tag{1}$$

- As a proxy to (1), authors replace $d_0 N_0 \sigma_1 = d_1 N_1 \sigma_0$ condition to

$$d_0 R_1\big(t^*(N_0/N_1)\big) = d_1 R_0\big(t^*(N_0/N_1)\big).$$

- After the class-wise quota selection, random pruning within each class is performed.

DRoP: Distributionally Robust Data Pruning (cont.)

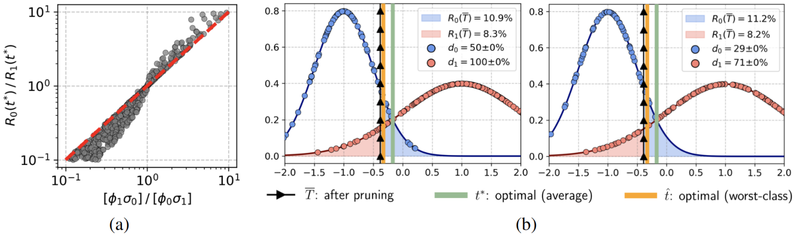


Figure 4: **(a)**: Class-wise risk ratios of the optimal solution $t^* = t^*(\phi_0/\phi_1)$ vs. optimal ratios based on Equation 5 computed for various $\sigma_0 < \sigma_1$ drawn uniformly from $[10^{-2}, 10^2]$ and $\phi_0 \sim U[0,1]$ and $\phi_1 = 1 - \phi_0$. The results are independent of μ_0, μ_1. **(b)**: Random pruning with DRoP. **Left:** $d = 75\%$; **Right:** $d = 50\%$.

- Class risks of the average and worst-class optimal decisions
  - $R_0\big(t^*(1)\big) = 4.8\%, R_1\big(t^*(1)\big) = 12.1\%$
  - $R_0(\hat{t}) = R_1(\hat{t}) = 9.1\%$

How About Other Data Pruning Methods in the Toy Example?

- Authors empirically and theoretically proved that a supervised variant of self-supervised pruning (SSP) [6] sticks to the average optimal solutions even after pruning.
  - Remove samples located within a certain margin $M > 0$ of each class mean.
  - Removes the easier class more aggressively.

How About Other Data Pruning Methods in the Toy Example? (cont.)

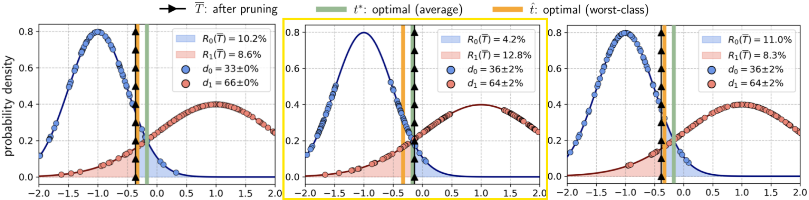


Figure 3: The effect of different pruning procedures on the solution mixture of Gaussians problem with $\mu_0 = -1$, $\mu_1 = 1$, $\sigma_0 = 0.5$, $\sigma_1 = 1$, and $\phi_0 = \phi_1$. Pruning to dataset density $d = 50\%$. **Left:** Random pruning with the optimal class-wise densities that satisfy $d_1\phi_1\sigma_0 = d_0\phi_0\sigma_1$. **Middle:** SSP. **Right:** Random pruning with respect to class ratios provided by the SSP algorithm. All results averaged across 10 datasets $\{D_i\}_{i=1}^{10}$ each with 400 points. The average ERM is $\overline{T} = \frac{1}{10}\sum_{i=1}^{10} T(D_i')$ fitted to pruned datasets D_i'. The class risks of the average and worst-class optimal decisions for this Gaussian mixture are $R_0[t^*(1)] = 4.8\%$, $R_1[t^*(1)] = 12.1\%$, and $R_0(\hat{t}) = R_1(\hat{t}) = 9.1\%$.

Proposed Algorithm and Experiments

DRoP for K-way Classification

- Input
  - `Dataset Density` d
  - Class sample size N_k for $k \in [K]$
    - ✓ $N = \sum_{k=1}^{K} N_k$
  - Validation accuracy r_k for $k \in [K]$
    - ✓ Evaluated given a query model ψ which is trained on a full dataset.

- Output: `Class Density` $d_k = d(1 - r_k)/Z$ for $k \in [K]$, where Z is a normalizing constant s.t.

$$dN = \sum_{k=1}^{K} d_k N_k.$$

DRoP for K-way Classification (cont.)

Algorithm 1: DRoP

Input: Target dataset density $d \in [0, 1]$.
For each class $k \in [K]$: original size N_k, validation recall $r_k \in [0, 1]$.

Initialize: Unsaturated set of classes $U \leftarrow [K]$, excess $E \leftarrow dN$, class densities $d_k \leftarrow 0 \ \forall k \in [K]$.

while $E > 0$ **do**
    $Z \leftarrow \frac{1}{E} \sum_{k \in U} N_k(1 - r_k)$;
    for $k \in U$ **do**
        $d'_k \leftarrow (1 - r_k)/Z$;
        $d_k \leftarrow d_k + d'_k$;
        $E \leftarrow E - N_k d'_k$;
        if $d_k > 1$ **then**
            $U \leftarrow U \setminus \{k\}$;
            $E \leftarrow E + N_k(d_k - 1)$;
            $d_k \leftarrow 1$
        end
    end
end

Return: $\{d_k\}_{k=1}^{K}$.

Experimental Results

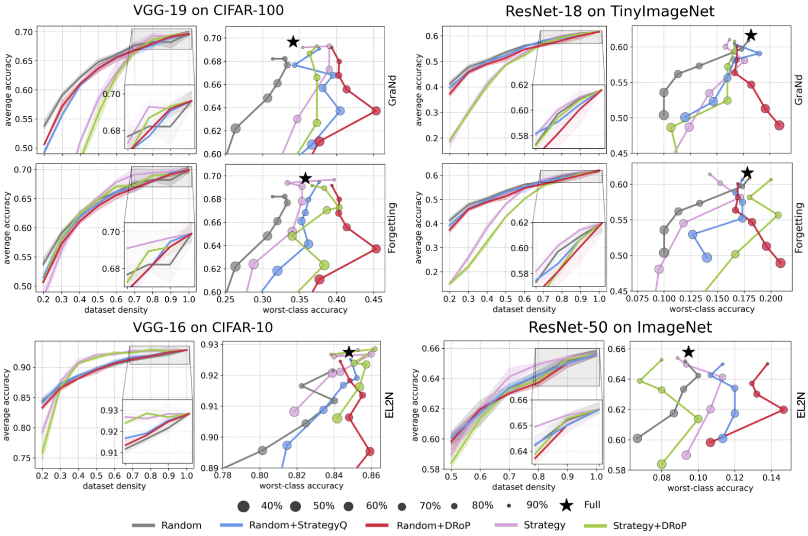


Figure 5: The average test performance of various data pruning protocols against dataset density and worst-class accuracy. All results averaged over 3 random seeds. Error bands represent min/max.

Experimental Results (cont.)

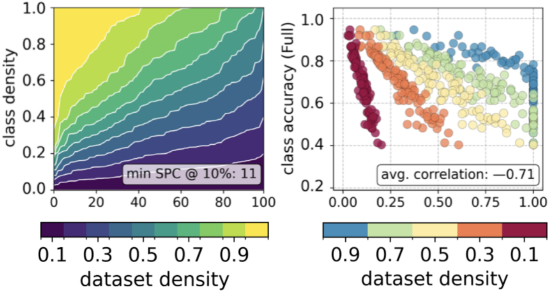


Figure 6: DRoP. **Left:** Sorted class densities at different dataset density levels. We report the minimum number of samples per class (SPC) at 10% dataset density. **Right:** Full dataset test class-wise accuracy against dataset density. We also report the correlation coefficient between these two quantities across classes, averaged over 5 dataset densities.

Experimental Results: Imbalanced Dataset

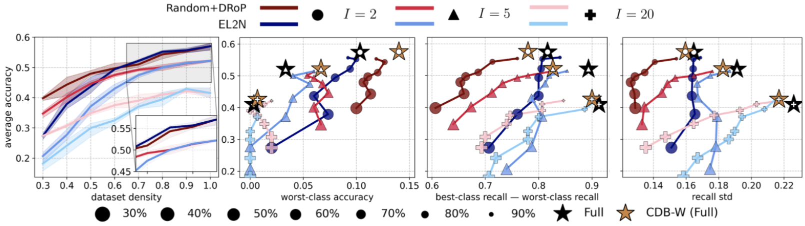


Figure 8: The average test performance of Random+DRoP (red-toned curves) and EL2N (blue-toned curves) against dataset density and measures of class robustness across dataset imbalance factors $I = 2, 5, 20$. ResNet-18 on imbalanced TinyImageNet. Results averaged over 3 random seeds. Error bands represent min/max.

Experimental Results: Spurious Correlation

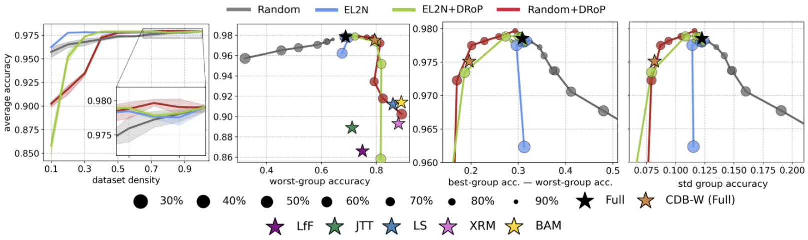


Figure 9: The average test performance of data pruning protocols and existing baselines against measures of group-wise robustness (ResNet-50 on Waterbirds). The results of data pruning and CDB-W averaged over 3 random seeds. Error bands represent min/max. To conform with Sagawa* et al. (2020), for this dataset, we compute average accuracy as a sum of group accuracies weighted by the original training group proportions. This explains the sharp degradation of the average performance of DRoP-backed pruning at low densities ($d \leq 0.4$): these datasets are skewed towards minority groups that weigh much less than severely pruned majority groups.

Limitations

- A Gap between the proposed algorithm and corresponding theoretical guarantees
- Cherry-picked experimental results

References

[1] Mariya Toneva et al. "An Empirical Study of Example Forgetting During Deep Neural Network Learning". In: *International Conference on Learning Representations* (2019).

[2] Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. "Deep Learning on a Data Diet: Finding Important Examples Early in Training". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 20596–20607.

[3] Muyang He et al. "Large-scale Dataset Pruning with Dynamic Uncertainty". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 7713–7722.

[4] Saptarshi Sinha, Hiroki Ohashi, and Katsuyuki Nakamura. "Class-difficulty Based Methods for Long-tailed Visual Recognition". In: *International Journal of Computer Vision* 130.10 (2022), pp. 2517–2531.

[5] Shiori Sagawa et al. "Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-case Generalization". In: *International Conference on Learning Representations* (2020).

[6] Ben Sorscher et al. "Beyond Neural Scaling Laws: Beating Power Law Scaling via Data Pruning". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 19523–19536.

Thank You!