Probabilistic Bilevel Coreset Selection

Xiao Zhou[1*], Renjie PI[1*], Weizhong Zhang[1*], Yong Lin[2], Tong Zhang[1,2]

[1]Hong Kong University of Science and Technology

[2]Google Research

Coreset Selection

- Though superior performances have been achieved via learning from huge amounts of data, the data-driven paradigm also poses several new challenges:
    - 1) the cumbersome dataset becomes harder to store and transfer;
    - 2) for some real applications, such as continual learning, one can only access a small number of training data at each stage of training;
    - 3) in some more extreme scenarios, where the training data is incorrectly labelled, or they are collected from different domains, more training data may even hurt the model's performance.

    Therefore, how to select a small subset (i.e coreset) comprised of most informative training samples, such that training on this subset can achieve comparable or even better performance with that on the full dataset becomes interesting.

Limitations of Previous works

- Traditional methods usually formulate the problem using uniform function approximation, which aims to find the subset from the entire set satisfying:

$$|\mathcal{L}(\boldsymbol{\theta}) - \hat{\mathcal{L}}(\boldsymbol{\theta})| \leq \epsilon \mathcal{L}(\boldsymbol{\theta}), \text{ for any } \boldsymbol{\theta} \in \mathbb{R}^p.$$

  - Where the first term on the left is the loss calculated over the entire dataset, and the second term is the loss calculated over the selected subset.

  - These methods cannot be applied to DNNs directly. The reason is that as DNNs are always highly nonconvex and the hypothesis set is significantly larger than traditional models, to obtain small uniform approximation error, one has to select a very large coreset.
- Recently, a more reasonable formulation based on bilevel optimization has been proposed:

$$\min_{\hat{\mathcal{W}}, \hat{\mathcal{D}} \subset \mathcal{D}, |\hat{\mathcal{D}}| \leq K} \mathcal{L}(\boldsymbol{\theta}^*(\mathcal{W}, \hat{\mathcal{D}}))$$
$$s.t. \ \boldsymbol{\theta}^*(\hat{\mathcal{W}}, \hat{\mathcal{D}}) = \arg\min_{\boldsymbol{\theta}} \hat{\mathcal{L}}(\boldsymbol{\theta}).$$

In this formulation, we only take into account the performance of the model trained on the coreset, i.e., the optimum , instead of achieving small approximation error for the loss function in the whole parameter space.

Probabilistic Bilevel Coreset Selection

- Our formulation for coreset selection:

$$\min_{\boldsymbol{s}\in\mathcal{C}} \Phi(\boldsymbol{s}) = \mathbb{E}_{p(\boldsymbol{m}|\boldsymbol{s})}\, \mathcal{L}(\boldsymbol{\theta}^*(\boldsymbol{m})),$$
$$s.t.\ \boldsymbol{\theta}^*(\boldsymbol{m}) \in \arg\min_{\boldsymbol{\theta}} \hat{\mathcal{L}}(\boldsymbol{\theta}; \boldsymbol{m})$$

$$\text{where } \mathcal{C} = \{\boldsymbol{s} : 0 \preceq \boldsymbol{s} \preceq 1, \|\boldsymbol{s}\|_1 \leq K\} \text{ is the domain.}$$

- Our formulation enjoys the following properties:
  - Our constraint C induces sparsity on the probability score, making most components of the optimal s either 0 or 1. That is, our finally learned stochastic mask is nearly deterministic
  - Due to our sparsity constraint, the selected coreset size of the inner loop is always small, which makes the optimization of model parameters very efficient.
  - Our outer objective is differentiable, allowing us to use general gradient based methods for optimization.
  - Our algorithm gradually improves the quality of the selected subset by explores the training set globally. Unlike greedy methods which make decisions at early stage and cannot remove redundant data once added to the coreset.

Optimization

- We use Policy Gradient Estimator (PGE), which calculates the gradient using forward instead of backward propagation. Our key idea can be illustrated by the following equations:

$$
\begin{aligned}
\nabla_{\boldsymbol{s}} \Phi(\boldsymbol{s}) =& \nabla_{\boldsymbol{s}} \mathbb{E}_{p(\boldsymbol{m}|\boldsymbol{s})} \mathcal{L}\left(\boldsymbol{\theta}^*(\boldsymbol{m})\right) \\
=& \nabla_{\boldsymbol{s}} \int \mathcal{L}\left(\boldsymbol{\theta}^*(\boldsymbol{m})\right) p(\boldsymbol{m}|\boldsymbol{s}) d\boldsymbol{m} \\
=& \int \mathcal{L}\left(\boldsymbol{\theta}^*(\boldsymbol{m})\right) \frac{\nabla_{\boldsymbol{s}} p(\boldsymbol{m}|\boldsymbol{s})}{p(\boldsymbol{m}|\boldsymbol{s})} p(\boldsymbol{m}|\boldsymbol{s}) d\boldsymbol{m} \\
=& \int \mathcal{L}\left(\boldsymbol{\theta}^*(\boldsymbol{m})\right) \nabla_{\boldsymbol{s}} \ln p(\boldsymbol{m}|\boldsymbol{s}) p(\boldsymbol{m}|\boldsymbol{s}) d\boldsymbol{m} \\
=& \mathbb{E}_{p(\boldsymbol{m}|\boldsymbol{s})} \mathcal{L}\left(\boldsymbol{\theta}^*(\boldsymbol{m})\right) \nabla_{\boldsymbol{s}} \ln p(\boldsymbol{m}|\boldsymbol{s}).
\end{aligned}
$$

- With PGE, we need only the loss value to calculate the gradient, which prevents deriving the second order term for implicit gradient. Thus greatly boosts the efficiency.

Experiments

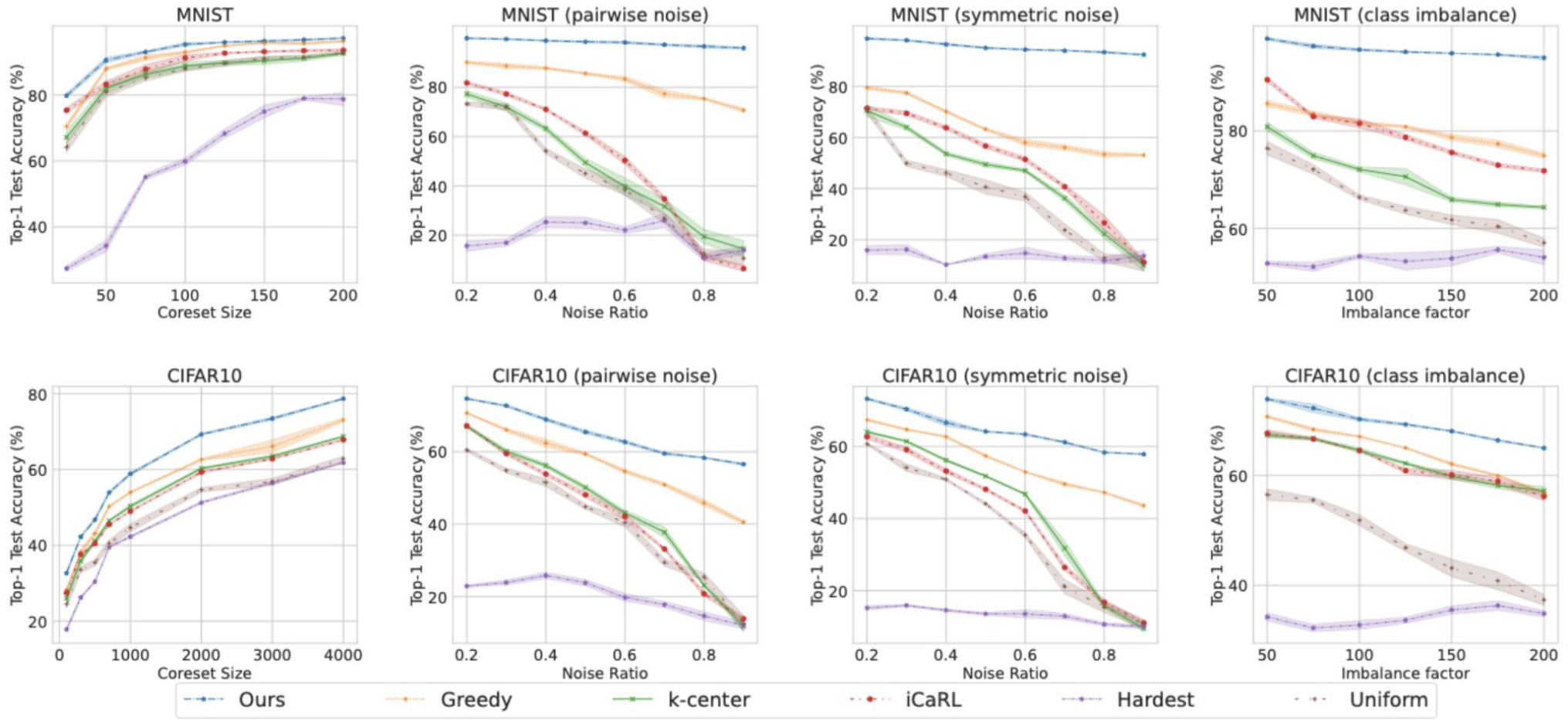

Figure 1. Performance comparison between our method and other baselines on data summarization task with various coreset sizes and different scenarios. For experiments with label noise and class imbalance, the coreset size is set to 1000 for MNIST and 5000 for CIFAR10. Our method consistently surpasses other baselines by a large margin. Notably, the performance of our method is stable even under challenging settings, while other methods begin to fail significantly.

Feature Selection

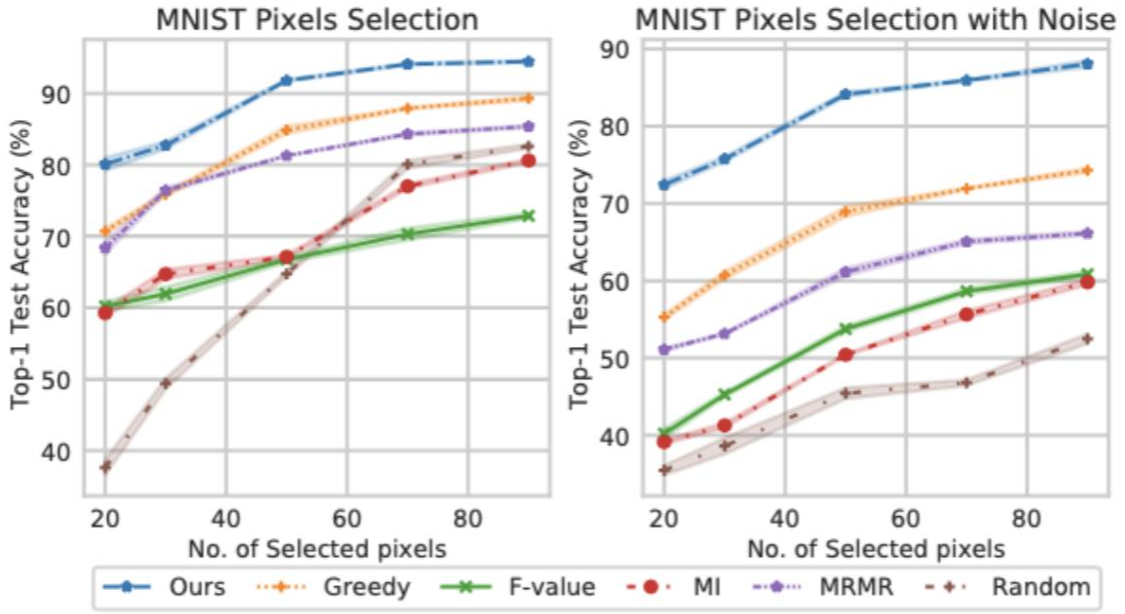

Figure 2. Left: We select a subset of pixel locations, then retain the corresponding pixels of all the training data for training and inference. Right: Pixel selection on dataset with gaussian noise applied to the images. Our method consistently surpasses other baselines by a large margin given different subset size constraints and the advantage is more evident with gaussian noise.

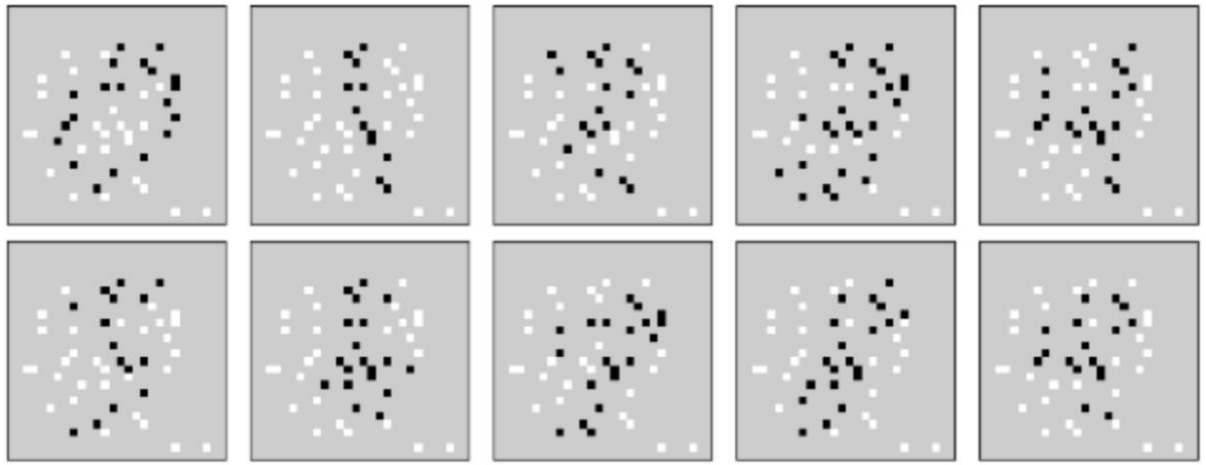

Figure 3. Visualization of selected pixels by our algorithm, where the numbers from 0 to 9 are demonstrated. The white pixels are selected by our algorithm and the black dots are those overlapped with the digits. We can see that the selected pixel locations effectively capture the important information of the images.

Continual learning

Table 1. Experiment result on continual learning for PermMNIST, SplitMNIST and CIFAR10 datasets. Normal stands for the standard dataset without modification; Noise represents symmetric label noise with 20% corruption ratio and Imbalance means the data has an class imbalance factor of 50. As demonstrated, using our proposed approach to construct the replay memory consistently surpasses other methods. Remarkably, the advantage of our global search strategy against the greedy counterpart becomes more prominent on challenging tasks: for tasks with label noise, our method surpasses the greedy counterpart by around 10%.

Datasets	PermMNIST			SplitMNIST			SplitCIFAR-10		
	Normal	Noise	Imbalance	Normal	Noise	Imbalance	Normal	Noise	Imbalance
Uniform	78.46	32.12	43.70	93.70	44.32	53.32	36.20	15.80	20.50
k-center embeddings	78.57	37.53	54.53	94.55	51.89	71.50	36.91	16.68	21.44
Hardest samples	76.79	15.38	38.72	91.57	17.23	56.39	28.10	9.63	15.63
iCaRL	79.68	39.36	67.53	95.13	60.99	72.23	34.52	18.47	25.47
Greedy Coreset	79.26	65.84	68.67	96.50	81.75	84.58	37.60	24.23	31.28
Ours	**80.60**	**74.26**	**75.32**	**98.15**	**92.23**	**94.30**	**39.10**	**31.20**	**35.30**

Ablation Analysis

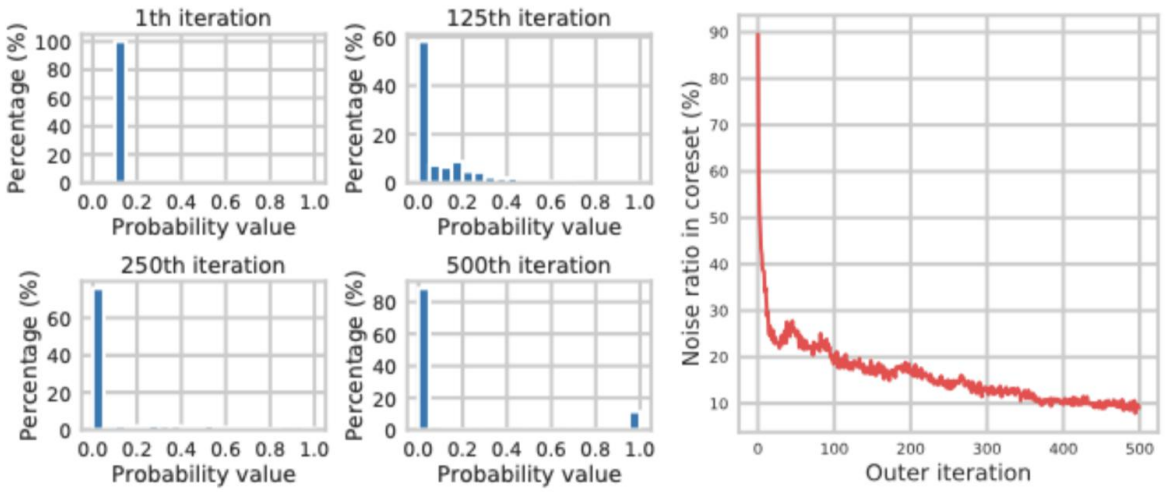

Figure 5. Left: The distribution of probability scores during the search. As the search progresses, most of the score values converge to either 0 or 1, which eventually renders a deterministic coreset with low variance. Right: the noise ratio in the selected coreset as the outer iteration increases. Our method progressively improves the quality of the coreset by learning the global information and updating the probability distribution accordingly.

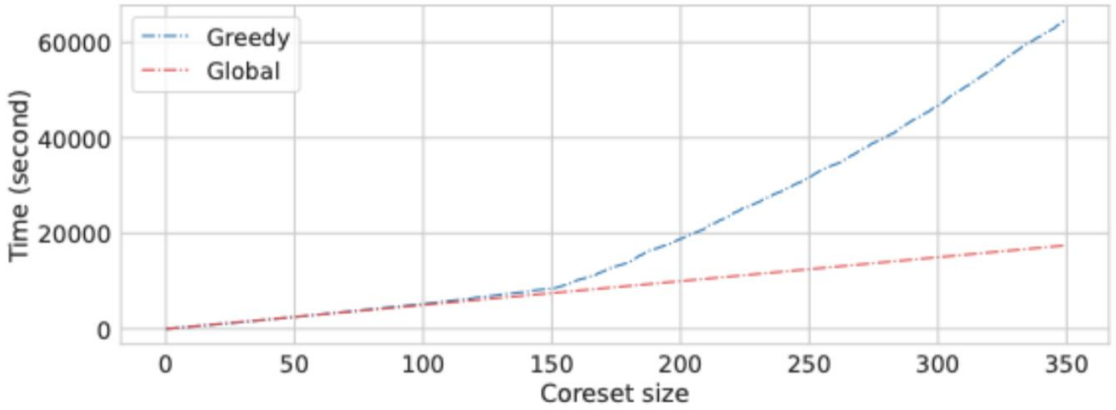

Figure 6. Comparison of time consumption between our method and the greedy counterpart. As the greedy coreset selection method needs to solve a bilevel problem for every newly added sample, the cost increases rapidly with the coreset size.

Thank You