Dataset Distillation for Audio Classification: A Data-Efficient Alternative to Active Learning

<u>Gautham Krishna Gudur</u>, Prof. Edison Thomaz

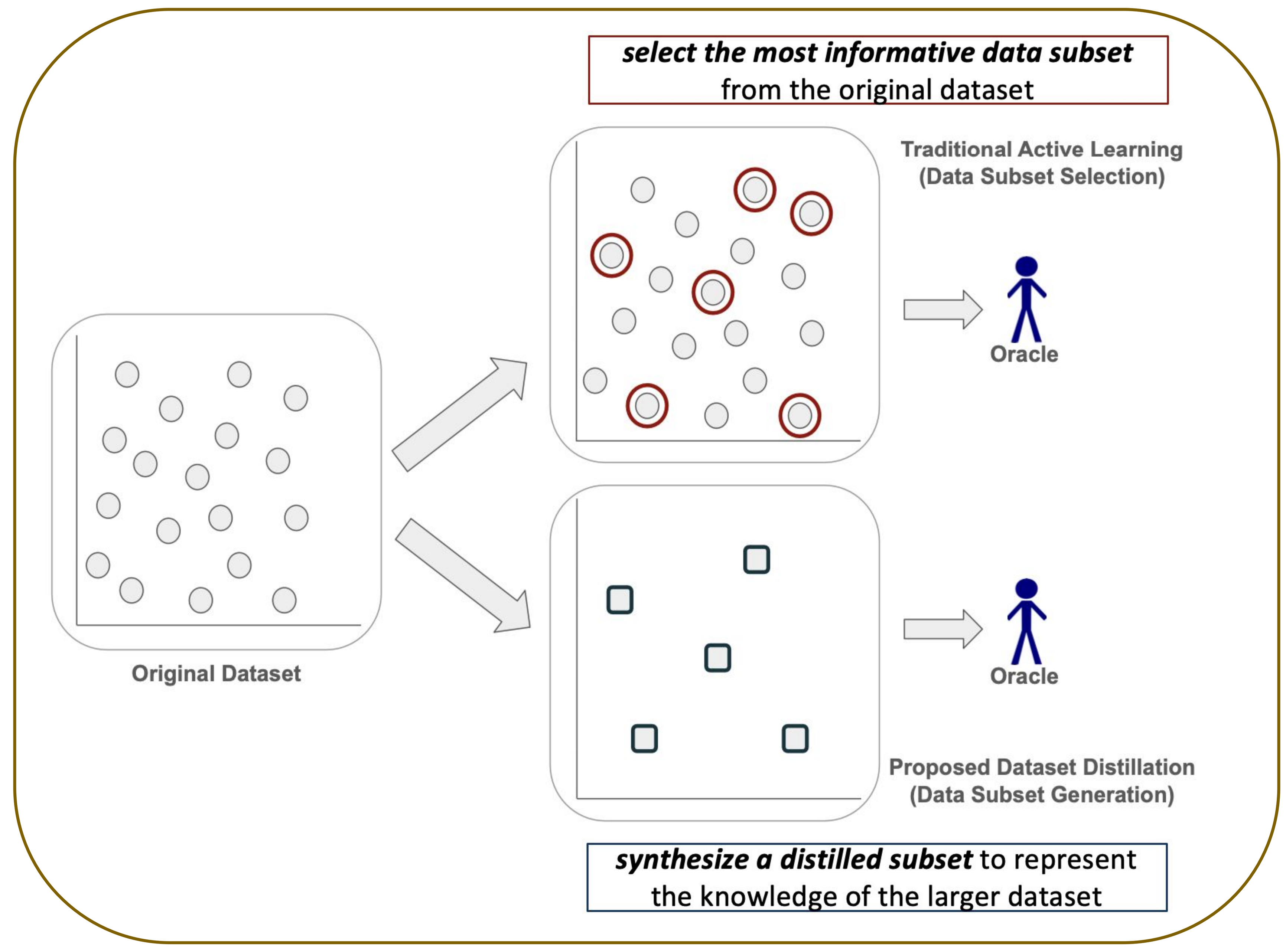
The University of Texas at Austin

SCAN ME

Problem Statement and Motivation

- Audio classification often require large labeled datasets. Problems –
  - ➔ Computationally expensive to train
  - ➔ Storage demands on resource-constrained devices
- **Active Learning**: reduce labeling efforts by selecting the most informative samples
  Problem – still requires thousands of audio segments from oracle (user)

What if we use Dataset Distillation (DD) as an alternative strategy to active learning?



Proposed Data Distillation Approach

- Synthesize *compact, high-fidelity data summaries* to reduce labeled data requirements for audio classification
- We use the RFAD method which employs random feature approximation, with principles from Neural network Gaussian processes (NNGP) and kernel regression
- Baseline active learning acquisition functions –
  – Max Entropy – Variation Ratios – Random
  – Bayesian Active Learning by Disagreement (BALD)

Algorithm 1 Our Proposed Approach

Input. Training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, Initial coreset $\mathcal{C} = \{(x'_i, y'_i)\}_{i=1}^{M}$, Number of random networks N, Output dimension of random networks M, Regularization parameter λ, Learning rate η

while not converged **do**
  Sample a batch $\mathcal{B} \subset \mathcal{D}$
  Initialize N random neural networks $\{f_{\theta_i}\}_{i=1}^{N}$
  for each $x \in \mathcal{B}$ **do**
    Compute random features $\Phi(x)$
  end for
  for each $x' \in \mathcal{C}$ **do**
    Compute random features $\Phi(x')$
  end for
  Compute kernel matrices $K_{\mathcal{BC}}$ and $K_{\mathcal{CC}}$
  Calculate predicted labels for the batch: $\hat{y}_{\mathcal{B}} = K_{\mathcal{BC}}(K_{\mathcal{CC}} + \lambda I)^{-1} y_{\mathcal{C}}$
  Compute loss: $\mathcal{L} = \|y_{\mathcal{B}} - \hat{y}_{\mathcal{B}}\|^2$
  Update coreset using gradient descent: $\mathcal{C} \leftarrow \mathcal{C} - \eta \nabla_{\mathcal{C}} \mathcal{L}$
end while
Ensure: Distilled coreset $\mathcal{C}$

Experiments and Results

Table 1: Comparison of test classification accuracy vs % of training samples for baseline methods and our proposed approach across all three datasets using a ResNet-18 model.

Method	Google Speech Commands		UrbanSound8K		ESC-50	
	% Samples	Accuracy	% Samples	Accuracy	% Samples	Accuracy
Total Training Data	100%	89.92	100%	79.27	100%	69.36
Max Entropy	60%	72.2	60%	62.15	60%	50.2
	40%	69.6	40%	57.45	40%	45.12
	30%	65.1	30%	53.36	30%	41.25
	20%	57.25	20%	45.18	20%	36.75
	0.029%	9.15	0.063%	7.56	0.15%	5.12
Variation Ratios	**60%**	**73.36**	60%	63.02	**60%**	**51.25**
	40%	70.85	40%	58.75	40%	45.36
	30%	65.3	30%	54.12	30%	41.58
	20%	58.72	20%	46.78	20%	36.24
	0.029%	9.24	0.063%	7.15	0.15%	5.84
BALD	60%	73.15	**60%**	**63.12**	60%	50.95
	40%	70.5	40%	58.95	40%	44.78
	30%	65.1	30%	54.08	30%	40.75
	20%	58.55	20%	46.42	20%	35.16
	0.029%	9.38	0.063%	7.39	0.15%	5.5
Random	60%	73.15	60%	62.87	60%	50.67
	40%	70.25	40%	59.08	40%	44.95
	30%	65.36	30%	54.45	30%	39.25
	20%	58.48	20%	46.92	20%	33.18
	0.029%	9.27	0.063%	7.72	0.15%	4.95
Proposed Method	**0.029%**	**72.24**	**0.063%**	**61.67**	**0.15%**	**49.65**
	0.017%	61.13	0.038%	50.24	0.09%	31.25
	0.012%	51.68	0.025%	37.85	0.0625%	17.96

Table 2: Comparison of test classification accuracy vs % of training samples for baseline methods and our proposed approach across all three datasets using a 4-layer CNN model.

Method	Google Speech Commands		UrbanSound8K		ESC-50	
	% Samples	Accuracy	% Samples	Accuracy	% Samples	Accuracy
Total Training Data	100%	87.45	100%	77.24	100%	67.62
Max Entropy	60%	69.92	60%	59.36	**60%**	**48.56**
	40%	66.25	40%	53.15	40%	44.78
	30%	63.55	30%	50.65	30%	39.05
	20%	56.18	20%	43.55	20%	34.56
	0.029%	8.27	0.063%	6.25	0.15%	4.85
Variation Ratios	60%	70.48	**60%**	**59.75**	60%	48.21
	40%	66.95	40%	53.5	40%	44.35
	30%	63.15	30%	49.87	30%	38.67
	20%	55.86	20%	43.78	20%	34.95
	0.029%	8.75	0.063%	5.92	0.15%	4.72
BALD	60%	70.27	60%	59.25	60%	47.75
	40%	66.18	40%	53.15	40%	43.85
	30%	62.67	30%	49.33	30%	38.75
	20%	56.25	20%	42.95	20%	33.48
	0.029%	8.35	0.063%	6.04	0.15%	4.3
Random	**60%**	**70.67**	60%	59.27	60%	48.05
	40%	67.05	40%	52.92	40%	44.72
	30%	62.18	30%	49.45	30%	38.02
	20%	56.67	20%	43.15	20%	35.05
	0.029%	8.96	0.063%	6.36	0.15%	4.67
Proposed Method	**0.029%**	**69.18**	**0.063%**	**58.52**	**0.15%**	**46.92**
	0.017%	57.45	0.038%	48.15	0.09%	28.05
	0.012%	45.67	0.025%	35.75	0.0625%	15.67

Number of Audio Samples per Class (AS/C)

Google Speech Commands		UrbanSound8K		ESC-50	
% Samples	AS/C	% Samples	AS/C	% Samples	AS/C
0.029%	50	0.063%	50	0.15%	5
0.017%	30	0.038%	30	0.09%	3
0.012%	20	0.025%	20	0.0625%	2

Upto ~3000x reduction in audio samples while offering competitive performance

Few other DD methods like –
- Data Condensation with Gradient Matching
- Differentiable Siamese Augmentation