Promises and Pitfalls of Threshold-based Auto-labeling

Harit Vishwakarma
hvishwakarma@cs.wisc.edu
Ph.D. Student
Dept. of Computer Sciences
University of Wisconsin-Madison

Frederic Sala
fredsala@cs.wisc.edu

Ramya Korlakai Vinayak
ramya@ece.wisc.edu

Heguang Lin
hmlin@seas.upenn.edu

Frederic Sala
fredsala@cs.wisc.edu

Ramya Korlakai Vinayak
ramya@ece.wisc.edu
ML needs labeled data and often a lot of it!

**Classical Supervised Learning**

- Diagnosing a novel disease using brain scans

- Labeled data

- Supervised Training

**Fine-tuning Foundation models or Aligning LLMs**

- Labeled data

- Web scale Unlabeled data

- Unsupervised Pre-training

- Supervised Fine-tuning

- Fine-tuned model
Getting labeled data is **costly and time-consuming**

Crowdsourcing is widely used to get labels

Takes a lot of time and money to get labels.

Took multiple years and a lot of human effort

14M Images, 20K Classes.

A screenshot of the ImageNet database online
How do we get accurately labeled data, while spending less time and money?
Automatically label datasets with minimal human feedback

Get labels for “minimal” points from human

Train a model on these labeled points and

Auto-label using the model
Auto-labeling systems are widely used

Auto-labeling Platforms

Auto-labeling is heavily used commercially.

Even in high risk applications

health care, telecom, recruiting...
Despite wide adoption, our understanding of auto-labeling systems is limited!
Despite wide adoption, our understanding of auto-labeling systems is limited!

To address this gap we develop a theoretical understanding of auto-labeling systems.
Auto-Labeling Errors and Their Impact

1. The output dataset may have labeling errors

2. The impact of errors in datasets is more severe
   a) Multiple downstream applications
   b) Longer shelf-life than models.
Quality and Quantity of Auto-labeled Data

Quantity

Auto-labeling Coverage

$\hat{P} = \frac{N_a}{N}$

Good Stuff maximize this

Quality

Auto-labeling Error

$\hat{\mathcal{E}} = \frac{M_a}{N_a}$

Bad Stuff minimize this

$N$ Number of unlabeled points

$A$ Set of auto-labeled points

$N_a$ Number of auto-labeled points

$M_a$ Number of labeling mistakes
Threshold-based Auto-labeling Workflow (TBAL)

**Input**
- Unlabeled Data i.i.d. from space \( \mathcal{X} \)
- Auto-labeling error tolerance \( \epsilon_a \)

**Output**
- Labeled Data

**0. Bootstrap**
- Create validation and initial training sets

**1. Learn a model** \( w \) using training set

**2. Find auto-labeling region, where the model can be trusted**

**3. Auto-label points in the identified region**

**4. Remove points in auto-labeling region**

**5. Get more human-labeled data for training and go to step 1**

---

**Model Class**
- \( \mathcal{H} : \mathcal{X} \rightarrow \mathcal{Y} \)
- \( h(x; w) = \text{sign}(w^T x) \)

**Empirical Risk Minimizer**
- \( g : \mathcal{X} \rightarrow T \subseteq \mathbb{R}^+ \)
- \( g(x; w) = |w^T x| \)

**Unlabeled Data**
- Validation Data

---

**Unlabeled Data**
- Input

**Labeled Data**
- Output
Step 2: Finding the Auto-labeling Region

Use the **validation data** to find the region where the classifier can be trusted

- Incorrect
- Correct

---

[Diagram showing the classification regions with validation data points and classifiers' decision boundaries]
Step 2: Finding the Auto-labeling Region

Estimate auto-labeling errors at several thresholds for each class separately.

Pick the smallest threshold giving error at most $\epsilon$.
Theoretical Results

Conditions on the validation data for accurate auto-labeling

In the general setup: No assumptions on data distribution and function classes

Upper bound on excess auto-labeling error

\[ O \left( \frac{1}{\sqrt{N_v}} + \mathfrak{R}_{N_v}(\mathcal{H}^{T,g}) \right) \]

\( \mathcal{H}^{T,g} := \mathcal{H} \times T \) \((h, t) \in \mathcal{H}^{T,g}\)

\((h, t)(x) := \begin{cases} h(x) & \text{if } g(h, x) \geq t \\ \text{abstain} & \text{o.w.} \end{cases} \)

Lower bound of \( \Omega \left( \frac{1}{\epsilon_a^2} \right) \) on number of validation samples to ensure auto-labeling error is below \( \epsilon_a \)
We validate the results empirically

Fix the auto-labeling error tolerance and the max number of training points algorithm can use.

### Vary the number of validation points

#### Unit ball (Synthetic)

| $N_v$ | Error (%) | Coverage (%) |
|-------|-----------|--------------|
|       | TBAL | AL+SC | TBAL | AL+SC |
| 100   | 3.10 ±1.30 | 0.68 ±0.81 | 71.43 ±8.86 | 96.95 ±1.01 |
| 400   | 1.65 ±0.65 | 0.32 ±0.15 | 93.27 ±2.59 | 96.91 ±0.99 |
| 800   | 1.08 ±0.47 | 0.24 ±0.16 | 96.01 ±1.16 | 96.31 ±1.36 |
| 1200  | 0.78 ±0.27 | 0.17 ±0.11 | 96.82 ±0.84 | 95.96 ±1.40 |
| 1600  | 0.65 ±0.20 | 0.13 ±0.08 | 96.93 ±0.57 | 95.70 ±1.38 |
| 2000  | 0.54 ±0.16 | 0.21 ±0.11 | 97.23 ±0.42 | 96.36 ±1.13 |

*# Classes = 2  \( \epsilon_A = 1\% \)*

*Max # training points = 500*

---

#### IMDB

| $N_v$ | Error (%) | Coverage (%) |
|-------|-----------|--------------|
|       | TBAL | AL+SC | TBAL | AL+SC |
| 200   | 2.28 ±0.21 | 3.11 ±0.86 | 68.24 ±6.20 | 57.77 ±13.09 |
| 400   | 1.29 ±0.10 | 1.98 ±0.40 | 63.81 ±4.86 | 63.06 ±10.70 |
| 600   | 1.41 ±0.20 | 1.81 ±0.22 | 69.64 ±3.98 | 62.92 ±9.20 |
| 800   | 1.62 ±0.30 | 2.04 ±0.35 | 67.45 ±3.72 | 63.22 ±7.89 |
| 1000  | 1.64 ±0.23 | 1.97 ±0.26 | 70.28 ±2.82 | 66.11 ±8.00 |

*# Classes = 2  \( \epsilon_A = 5\% \)*

*Max # training points = 500*

---

#### Tiny Imagenet

| $N_v$ | Error (%) | Coverage (%) |
|-------|-----------|--------------|
|       | TBAL | AL+SC | TBAL | AL+SC |
| 2000  | 0.0 ±0.0 | 0.0 ±0.0 | 0.0 ±0.0 | 0.0 ±0.0 |
| 4000  | 10.50 ±6.01 | 7.37 ±4.57 | 0.47 ±0.05 | 0.48 ±0.06 |
| 6000  | 10.61 ±6.82 | 7.71 ±1.03 | 10.16 ±1.10 | 4.31 ±1.10 |
| 8000  | 9.90 ±0.63 | 6.80 ±0.77 | 25.84 ±1.57 | 14.43 ±2.01 |
| 10000 | 8.97 ±0.36 | 6.87 ±0.48 | 32.19 ±1.52 | 21.96 ±1.35 |

*# Classes = 200  \( \epsilon_A = 10\% \)*

*Max # training points = 10000*

---

As expected, we observe

- Less validation data \( \rightarrow \) high auto-labeling errors and high variance in coverage
- Suff. Large validation data \( \rightarrow \) less auto-labeling errors and less variance in coverage
Summary and Takeaways

1. Auto labeling is a promising solution to obtain labeled data.

2. Our work develops a theoretical understanding of auto-labeling systems.

3. The promise — Seemingly bad models can auto-label significant portion of data with good accuracy.

4. The pitfall — Hidden downside is large amount validation data needed to ensure good accuracy.
Thank You

Checkout our paper and code!

Paper  https://openreview.net/pdf?id=RUCFAKNDb2
Code   https://github.com/harit7/TBAL-NeurIPS-23

Come to our poster @ NeurIPS

Hall B1 + B2 #1103
Wed 13 Dec
3 p.m. - 5 p.m. PST

Contact us

Frederic Sala
fredsala@cs.wisc.edu

Heguang Lin
hgl@seas.upenn.edu

Harit Vishwakarma
hvishwakarma@cs.wisc.edu

Ramya Korlakai Vinayak
ramya@ece.wisc.edu