Crowdsourced PAC Learning under Classification Noise

Shelby Heinecke (joint work with Lev Reyzin)

Department of Mathematics, Statistics, and Computer Science
University of Illinois - Chicago

Overview

- Introduction
- Algorithm
- Evaluation
- Extensions
- Conclusion

Introduction

Introduction and Motivation

- In supervised learning, labelled data is needed to train classifiers
- Crowdsourcing can be used to get labels, but has issues:
  - Crowd workers make mistakes
  - Labelled data not always available to evaluate crowd workers

Our Solution

We develop a probably approximately correct (PAC) algorithm that uses

1. majority-voting
2. pure-exploration bandits
3. noisy-PAC learning

to produce a trained classifier using only data points labeled by noisy crowd workers.

Our Results

- Bounds on the number of tasks assigned to workers (cost)
- Improvement over baseline approaches and related work
- Easily adapted to fit additional settings:
  - asymmetric noise
  - crowd worker task limits

Setting

- Dataset, X, **unlabeled**
- Set of crowd workers, W
- Each $w_i \in W$ has error rate $\eta_i \leq \frac{1}{2}$ (**unknown a priori**)

Goal

Assign to crowd workers as few tasks as possible to train a good classifier.

Related Work – Highlights

- Majority Voting in Crowdsourcing
  - Number of tasks needed for majority voting to give correct label in noisy setting: $O(\log(|X|))$
    - ★ larger $|X| \implies$ more labels per point needed!
  - **We use majority voting on small subset of X**
    - ★ Not too expensive
    - ★ Circumvent need for ground-truth data
  - More generally on label aggregation – lots of work but no end-to-end solutions
- PAC Learning in Crowdsourcing
  - Previous work is similar to our baseline approaches
  - Most related work, Awasthi et al. (2017), assumes:
    - ★ some ground truth labels are available
    - ★ some perfect performing crowd workers

Related Work – Highlights

Multi-Armed Bandit Problems

Given a series of “arms” to pull, the learner:

- decides which arms to pull
- knows only the rewards of their previous choices
- wants to **maximize total profit**

- Bandits in Crowdsourcing
  - task allocation, worker selection can be cast as a bandit problem
  - we build upon previous works
    - ★ **previous works do not focus on training a classifier**

Algorithm

Algorithm Overview

1. **Majority voting** to get a ground truth set
   - get accurately labeled data with probability $1 - \delta$
2. **Bandit algorithms** to identify good crowd workers
   - uses the "ground-truth" data from previous step
3. **Return model** that agrees with the labels from good crowd workers
   - number of labels needed from good workers outlined in noisy-PAC learning guarantee

Step 1: Majority Voting

Goal

Get accurate labels on T data points with probability $1-\delta$.

Number of tasks needed **per data point**:

# of points to be labeled by majority voting

$$O\left(\frac{\log(T/\delta)}{(1-2\bar{\eta}_W)^2}\right)$$

failure rate

average error of all workers

Proof.

$\Pr[\mathrm{MAJ}(x)) \neq c(x)] \leq 2e^{-n(1-2\bar{\eta}_W)^2/2}$ by Hoeffding inequality. Upper bound error of majority voting ($\leq \frac{\delta}{T}$). □

Step 2: Bandit Algorithms

Goal

Identify Δ-optimal crowd worker(s).

- Δ-optimal worker(s) – error rate is within Δ of the lowest error rate
- Use bandit algorithms in literature
  - our analysis uses vanilla setting
  - easily use more sophisticated bandit settings, such as sleeping bandits, exact best worker(s), etc.

Step 2: Bandit Algorithms

- Casting into bandit problem:
  - use ground-truth data from previous step
  - selected crowd worker gets data point
  - compute rewards
    - 1 – if worker is correct
    - 0 – otherwise
- State-of-the-art for vanilla Δ-optimal arm: OptMAI from Zhou et al. (2014).
  - identifies Δ-optimal worker(s) with probability $1 - \delta$

Step 2: Bandit Algorithms

Key Observation

The number of arm pulls needed to run bandit algorithms determines the number of ground-truth points T that we need from Step 1.

To identify Δ-optimal worker(s) with probability $1 - \delta$, the following number of tasks are assigned in total by OptMAI:

Δ-Optimal Worker:

$$O\left(\frac{n}{\Delta^2}\log\left(\frac{n}{\delta}\right)\right)$$

difference b/t error of worker identified and error of best worker

failure rate

Δ-Optimal Set of K Workers:

$$O\left(\frac{n}{\Delta^2}\left(1 + \frac{\log(\frac{1}{\delta})}{K}\right)\right)$$

difference b/t avg error of K workers identified and avg error of best K workers

Step 2: Bandit Algorithms

Lemma (Special Case)

Suppose there exists at least one perfect performing worker *in the crowd. Then,*

$$O(\frac{n}{\Delta}\log(\frac{n}{\delta}))$$

tasks are sufficient to identify a Δ-optimal worker with probability $1-\delta$.

Note: Dependence on Δ and not Δ^2

Proof.

A crowd worker with error Δ is observed to be perfect on t instances with probability $(1-\Delta)^t \leq e^{-\Delta t}$. Bound this by $\frac{\delta}{n}$ and solve for t. □

Step 3: Returning the Model via Noisy-PAC Learning

Goal

Return model that agrees with responses of the Δ-optimal worker(s).

- How many samples do we need from the Δ-optimal workers?
- Which model so we choose?

Noisy PAC Learning [Angluin and Laird 1987; Laird 1988]

For set of models C with finite VC-dimension d and any

- error rate $\epsilon > 0$
- confidence $\delta > 0$
- distribution D on the data X
- sample S drawn i.i.d. from $D \sim X$, **flip label with prob** $\eta < \frac{1}{2}$

where

$$|S| = O\left(\frac{d\log(\frac{1}{\delta})}{(1-2\eta)^2}\right),$$

an empirical risk minimizer (ERM) model, $h_{ERM} \in C$, satisfies the PAC criterion (i.e. h_{ERM} is probably approximately correct).

Key Observation

This formulation has one noise rate – **we have a crowd of noise rates**.

Step 3: Returning the Model via Noisy-PAC Learning

Adapting noisy PAC to determine # of tasks to assign:

- One Δ-optimal worker in bandits step:

$$\mathcal{O}\left(\frac{d\log(1/\delta)}{\epsilon(1-2(\bar{\eta}^{*}_{1,W}+\Delta))^2}\right)$$

Error rate of the **best worker**

- K Δ-optimal workers in bandits step:

$$\mathcal{O}\left(\frac{d\log(1/\delta)}{\epsilon(1-2(\bar{\eta}^{*}_{K,W}+\Delta))^2}\right)$$

Error rate of the **best K workers**

Final Output

$h_{ERM} \in C$ is the final model.

Total Task Complexity (Cost)

Satisfying PAC Criterion

Set the failure rate of each step to $\leq \frac{\delta}{3}$.

Summing the cost of all three steps:

- One Δ-optimal worker in bandits step:

$$\tilde{\mathcal{O}}\left(\frac{\log^2(n/\delta)}{(1-2\bar{\eta}^*_{1,W})^2(1-2\bar{\eta}_W)^2} + \frac{\left(n+\frac{d}{\epsilon}\right)\log(1/\delta)}{(1-2\bar{\eta}^*_{1,W})^2}\right)$$

  - $\bar{\eta}^*_{1,W}$ = error of the best worker

- K Δ-optimal workers in bandits step:

$$\tilde{\mathcal{O}}\left(\frac{n\log^2(1/\delta)}{(1-2\bar{\eta}^*_{K,W})^2(1-2\bar{\eta}_W)^2} + \frac{d\log(1/\delta)}{\epsilon(1-2\bar{\eta}^*_{K,W})^2}\right)$$

  - $\bar{\eta}^*_{K,W}$ = error of the best K workers

Evaluation

Baseline Approach #1

$\bar{\eta}_W$ = average error rate of crowd workers

1. Assign
$$\mathcal{O}\left(\frac{d\log(1/\delta)}{\epsilon(1-2\bar{\eta}_W)^2}\right) \tag{1}$$
distinct tasks to crowd workers, uniformly at random.
2. Return h_{ERM}

Proof.

This is a PAC algorithm due to [Angluin and Laird 1987; Laird 1988]. □

Total Tasks Assigned to Crowd:

$$\mathcal{O}\left(\frac{d\log(1/\delta)}{\epsilon(1-2\bar{\eta}_W)^2}\right)$$

Baseline Approach #2

1. Majority vote to get ground-truth set of size

$$\mathcal{O}(\frac{d\log(\frac{1}{\delta})}{\epsilon})$$

correct with probability $1-\delta$.

2. Return h_{ERM}

Proof.

PAC algorithm by classic result in PAC learning theory. □

Total Tasks Assigned to Crowd:

$$\tilde{\mathcal{O}}\left(\frac{d\log(1/\delta)}{\epsilon(1-2\bar{\eta}_W)^2}\right)$$

Comparison to Baselines

$$O\left(\frac{d\log(1/\delta)}{\epsilon(1-2\bar{\eta}_W)^2}\right)$$

1/ε multiplied by fraction dependent on **average error of all workers** in *W*

Bound #1: Using One Good Worker

$$\tilde{O}\left(\frac{\log^2(\frac{n}{\delta})}{(\frac{1}{2}-\bar{\eta}^*_{1,W})^2(\frac{1}{2}-\bar{\eta}_W)^2}+\frac{(n+\frac{d}{\epsilon})\log(\frac{1}{\delta})}{(\frac{1}{2}-\bar{\eta}^*_{1,W})^2}\right)$$

1/ε multiplied by fraction dependent on **error of best worker**

Bound #2: Using *K* Good Workers

$$\tilde{O}\left(\frac{\log^2(\frac{1}{\delta})}{(\frac{1}{2}-\bar{\eta}^*_{K,W})^2(\frac{1}{2}-\bar{\eta}_W)^2}+(n+\frac{d}{\epsilon})\frac{\log(\frac{1}{\delta})}{(\frac{1}{2}-\bar{\eta}^*_{K,W})^2}\right)$$

1/ε multiplied by fraction dependent on **average error of top *K* workers**

Key Observation

The term multiplied by the $\frac{d}{\epsilon}$ is most important since $\epsilon \to 0$ in practice.

Extensions

Extension #1: Asymmetric Classification Noise

- Workers can have different error rates for each class (Dawid and Skene, 1979)
  - $\eta_i^+, \eta_i^- < \frac{1}{2}$
- Modifications:
  1. Need
$$\mathcal{O}\left(\frac{\log(T/\delta)}{(1-2\max(\bar{\eta}_W^+,\bar{\eta}_W^-))^2}\right)$$
     tasks per point for majority voting to be accurate with probability 1-δ
  2. Noisy PAC learning requires
$$O\left(\frac{d\log(1/\delta)}{\epsilon(1-2\max(\eta^+,\eta^-))^2}\right)$$
     tasks so that h_{ERM} is PAC.
- Total tasks assigned is polynomial
- average one-sided errors → best one-sided errors

Extension #2: Worker Limits

- Suppose workers can perform at most $B > 0$ tasks
- Task limit determines the number of Δ-optimal workers, K, in bandit step
  - $K = \frac{\text{number of tasks for Noisy PAC}}{B - \text{number of tasks for OptMAI(K)}}$
  - Higher $B \implies$ smaller number of Δ-optimal workers
  - Lower $B \implies$ greater number of Δ-optimal workers

Conclusion

Conclusion

Our algorithm

- needs only unlabelled data
- produces a trained classifier as output
- uses majority voting, bandits, and noisy-PAC learning
- satisfies PAC learning criterion
- improves upon baseline approaches
- can be easily extended to a variety of crowdsourcing settings

Future work:

- evaluate experimentally
- extensions to other crowd sourcing settings