Vox Populi: Collecting High-Quality Labels from a Crowd

Authors: Ofer Dekel, Ohad Shamir (COLT 2009)
Presenter: Haofan Zhang

David R. Cheriton School of Computer Science,
University of Waterloo,
Waterloo, Canada

October 20, 2014

Agenda

- Traditional machine learning focuses on the single-teacher setting

- Traditional machine learning focuses on the single-teacher setting
- We are faced with the problem of learning from crowd

- Traditional machine learning focuses on the single-teacher setting
- We are faced with the problem of learning from crowd
- Therefore, we are interested in identifying and removing low-quality teachers

Several challenges:

Several challenges:

- No prior knowledge on the identity or the quality of the teacher

Several challenges:

- No prior knowledge on the identity or the quality of the teacher
- No access to gold-set of perfectly labeled examples

Several challenges:

- No prior knowledge on the identity or the quality of the teacher
- No access to gold-set of perfectly labeled examples
  - Moreover, a typical teacher only labels a handful set of examples

Several challenges:

- No prior knowledge on the identity or the quality of the teacher
- No access to gold-set of perfectly labeled examples
  - Moreover, a typical teacher only labels a handful set of examples
- No control on assignment of examples

Several challenges:

- No prior knowledge on the identity or the quality of the teacher
- No access to gold-set of perfectly labeled examples
  - Moreover, a typical teacher only labels a handful set of examples
- No control on assignment of examples
  - Prevent us from applying repeated labeling.
  - Even applicable, should be avoided because of the cost.

Ultimately, our problem is to:

- Work with raw labeled data, with single noisy label per example
- Detect and eliminate low-quality teaches in a principled and effective manner

Vox populi, vox Dei
–The voice of the people [is] the voice of God.

Suppose we have **multiple labels** for each example

Suppose we have multiple labels for each example

- If most of teaches are good, we can simply take the average or majority over repeated labels

Suppose we have multiple labels for each example

- If most of teaches are good, we can simply take the average or majority over repeated labels
- Then we treat this aggregated label as ground truth and count incorrect label provided by each teacher

Suppose we have multiple labels for each example

- If most of teaches are good, we can simply take the average or majority over repeated labels
- Then we treat this aggregated label as ground truth and count incorrect label provided by each teacher
- Once we identify low-quality teachers, we can ignore them in the future.

Suppose we have multiple labels for each example

- If most of teaches are good, we can simply take the average or majority over repeated labels
- Then we treat this aggregated label as ground truth and count incorrect label provided by each teacher
- Once we identify low-quality teachers, we can ignore them in the future.

However, we don't have aggregated labels...

Introduction
A Simple Algorithm

Suppose we have multiple labels for each example

- If most of teaches are good, we can simply take the average or majority over repeated labels
- Then we treat this aggregated label as ground truth and count incorrect label provided by each teacher
- Once we identify low-quality teachers, we can ignore them in the future.

However, we don't have aggregated labels...

We want to simulate them!

Simulating aggregated labels:

Simulating aggregated labels:

- Specifically, we train a hypothesis(classifier) on the entire unfiltered dataset

Simulating aggregated labels:

- Specifically, we train a hypothesis(classifier) on the entire unfiltered dataset
- Then we regard the predictions of this hypothesis as the ground truth.

Simulating aggregated labels:

- Specifically, we train a hypothesis(classifier) on the entire unfiltered dataset
- Then we regard the predictions of this hypothesis as the ground truth.
- We pretend that we can rely on it, and eliminate low-quality teachers!

Settings and Notations

We focus on binary classification setting:

- Instance space: $\mathcal{X} \subseteq \mathbb{R}^n$

We focus on binary classification setting:

- Instance space: $\mathcal{X} \subseteq \mathbb{R}^n$
- Test Probability Distribution: $\mathcal{D} : \mathcal{X} \times \{-1, +1\}$

Settings and Notations

We focus on binary classification setting:

- Instance space: $\mathcal{X} \subseteq \mathbb{R}^n$
- Test Probability Distribution: $\mathcal{D} : \mathcal{X} \times \{-1, +1\}$
- Given dataset: $S = \{\mathbf{x}_i, y_i\}_{i=1}^m$

Settings and Notations

We focus on binary classification setting:

- Instance space: $\mathcal{X} \subseteq \mathbb{R}^n$
- Test Probability Distribution: $\mathcal{D} : \mathcal{X} \times \{-1, +1\}$
- Given dataset: $S = \{\mathbf{x}_i, y_i\}_{i=1}^m$
- The ML algorithm minimizes:

$$\hat{F}_\lambda(\mathbf{w}, S) = \lambda ||\mathbf{w}||^2 + \frac{1}{m} \sum_{i=1}^{m} \ell(f(\mathbf{w}, \mathbf{x}_i), y_i)$$

- Additionally,

$$f(\mathbf{w}, \mathbf{x}_i) = \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle$$

represents application of classifier $\mathbf{w}$ to the instance $\mathbf{x}_i$

Settings and Notations

In typical supervised learning setting:

- we assume that a training set S is sampled i.i.d from $\mathcal{D}$

Settings and Notations

In typical supervised learning setting:

- we assume that a training set S is sampled i.i.d from $\mathcal{D}$

Here, we introduce an extra stage where data is labeled by a set of k teachers:

Settings and Notations

In typical supervised learning setting:

- we assume that a training set S is sampled i.i.d from $\mathcal{D}$

Here, we introduce an extra stage where data is labeled by a set of k teachers:

- There exists k classifiers $\{h_1(\mathbf{x}), h_2(\mathbf{x}), \cdots, h_k(\mathbf{x})\} : \mathcal{X} \to \{-1, +1\}$

Settings and Notations

In typical supervised learning setting:

- we assume that a training set S is sampled i.i.d from $\mathcal{D}$

Here, we introduce an extra stage where data is labeled by a set of k teachers:

- There exists k classifiers $\{h_1(\mathbf{x}), h_2(\mathbf{x}), \cdots, h_k(\mathbf{x})\} : \mathcal{X} \to \{-1, +1\}$
- For each unlabeled instance $\mathbf{x}$, we choose a teacher $t \in \{1, \cdots, k\}$ at random (uniformly here for simplicity)

Settings and Notations

In typical supervised learning setting:

- we assume that a training set S is sampled i.i.d from $\mathcal{D}$

Here, we introduce an extra stage where data is labeled by a set of k teachers:

- There exists k classifiers $\{h_1(\mathbf{x}), h_2(\mathbf{x}), \cdots, h_k(\mathbf{x})\} : \mathcal{X} \to \{-1, +1\}$
- For each unlabeled instance $\mathbf{x}$, we choose a teacher $t \in \{1, \cdots, k\}$ at random (uniformly here for simplicity)
- This results in splitting the sample into k subsets, $S_1, \cdots, S_k$

Settings and Notations

- This process can be viewed as sampling an unlabeled dataset and labeling it using $\bar{h}(\mathbf{x})$,

Settings and Notations

- This process can be viewed as sampling an unlabeled dataset and labeling it using $\bar{h}(\mathbf{x})$,
  where $\bar{h}(\mathbf{x})$ is the random classifier defined by randomly choosing a hypothesis from $h_1, \cdots, h_k$

Settings and Notations

- This process can be viewed as sampling an unlabeled dataset and labeling it using $\bar{h}(\mathbf{x})$,
  where $\bar{h}(\mathbf{x})$ is the random classifier defined by randomly choosing a hypothesis from $h_1, \cdots, h_k$
- Remeber we want to minimize $\hat{F}_\lambda(\mathbf{w}, S)$:

$$\hat{F}_\lambda(\mathbf{w}, S) = \lambda ||\mathbf{w}||^2 + \frac{1}{m} \sum_{i=1}^{m} \ell(f(\mathbf{w}, \mathbf{x}_i), y_i)$$

Settings and Notations

- This process can be viewed as sampling an unlabeled dataset and labeling it using $\bar{h}(\mathbf{x})$,
  where $\bar{h}(\mathbf{x})$ is the random classifier defined by randomly choosing a hypothesis from $h_1, \cdots, h_k$
- Remeber we want to minimize $\hat{F}_\lambda(\mathbf{w}, S)$:

$$\hat{F}_\lambda(\mathbf{w}, S) = \lambda||\mathbf{w}||^2 + \frac{1}{m}\sum_{i=1}^{m} \ell(f(\mathbf{w}, \mathbf{x}_i), y_i)$$

- Then, it can be seen as the empirical counterpart of minimizing

$$F_\lambda(\mathbf{w}) = \lambda||\mathbf{w}||^2 + \mathbb{E}\left[\ell(f(\mathbf{w}, \mathbf{x}), \bar{h}(\mathbf{x}))\right]$$

- We denote $\mathbf{w}^*$ as the minimizer of $F_\lambda(\mathbf{w})$

Settings and Notations

- Remember that, our goal is to identify and prune away low-quality teachers.

Settings and Notations

- Remember that, our goal is to identify and prune away low-quality teachers.
- After pruning, only a set of high-quality teachers are left

Settings and Notations

- Remember that, our goal is to identify and prune away low-quality teachers.
- After pruning, only a set of high-quality teachers are left
- We denote $\bar{h}_T(\cdot)$ as randomly pick one classifier from high-quality teachers

Settings and Notations

- Error rate of teacher t:

$$e_t = \Pr_{(\mathbf{x},y)\sim D}(y h_t(\mathbf{x}) < 0)$$

Settings and Notations

- Error rate of teacher t:

$$e_t = \Pr_{(\mathbf{x},y)\sim D}(yh_t(\mathbf{x}) < 0)$$

- Error rate of entire crowd before pruning:

$$\bar{e} = \Pr_{(\mathbf{x},y)\sim D}(y\bar{h}(\mathbf{x}) < 0)$$

Settings and Notations

- Error rate of teacher t:

$$e_t = \Pr_{(\mathbf{x},y)\sim D}(y h_t(\mathbf{x}) < 0)$$

- Error rate of entire crowd before pruning:

$$\bar{e} = \Pr_{(\mathbf{x},y)\sim D}(y \bar{h}(\mathbf{x}) < 0)$$

- Error rate of entire crowd after pruning:

$$\bar{e}_T = \Pr_{(\mathbf{x},y)\sim D}(y \bar{h}_T(\mathbf{x}) < 0 | S)$$

Settings and Notations

- Error rate of teacher t:

$$e_t = \Pr_{(\mathbf{x},y)\sim D}(yh_t(\mathbf{x}) < 0)$$

- Error rate of entire crowd before pruning:

$$\bar{e} = \Pr_{(\mathbf{x},y)\sim D}(y\bar{h}(\mathbf{x}) < 0)$$

- Error rate of entire crowd after pruning:

$$\bar{e}_T = \Pr_{(\mathbf{x},y)\sim D}(y\bar{h}_T(\mathbf{x}) < 0|S)$$

However, we don't know $\mathcal{D}$ nor h_t, we cannot calculate e_t directly!

Settings and Notations

- Error rate of teacher t:

$$e_t = \Pr_{(\mathbf{x},y)\sim D}(yh_t(\mathbf{x}) < 0)$$

- Error rate of entire crowd before pruning:

$$\bar{e} = \Pr_{(\mathbf{x},y)\sim D}(y\bar{h}(\mathbf{x}) < 0)$$

- Error rate of entire crowd after pruning:

$$\bar{e}_T = \Pr_{(\mathbf{x},y)\sim D}(y\bar{h}_T(\mathbf{x}) < 0|S)$$

However, we don't know $\mathcal{D}$ nor h_t, we cannot calculate e_t directly! We need to look at something different!

Settings and Notations

The true “error-rate” according to $\mathcal{D}$:

$$e_t = \Pr_{(\mathbf{x},y)\sim D}(yh_t(\mathbf{x}) < 0)$$

$$\bar{e} = \Pr_{(\mathbf{x},y)\sim D}(y\bar{h}(\mathbf{x}) < 0)$$

$$\bar{e}_T = \Pr_{(\mathbf{x},y)\sim D}(y\bar{h}_T(\mathbf{x}) < 0|S)$$

Settings and Notations

The true "error-rate" according to $\mathcal{D}$:

$$e_t = \Pr_{(\mathbf{x},y)\sim D}(yh_t(\mathbf{x}) < 0)$$
$$\bar{e} = \Pr_{(\mathbf{x},y)\sim D}(y\bar{h}(\mathbf{x}) < 0)$$
$$\bar{e}_T = \Pr_{(\mathbf{x},y)\sim D}(y\bar{h}_T(\mathbf{x}) < 0|S)$$

The idea is to look at the "error-rate" with respect to $\mathbf{w}^*$:

$$\epsilon_t = \Pr(h_t(\mathbf{x})f(\mathbf{w}^*, \mathbf{x}) < 0)$$
$$\bar{\epsilon} = \Pr(\bar{h}(\mathbf{x})f(\mathbf{w}^*, \mathbf{x}) < 0)$$
$$\bar{\epsilon}_T = \Pr(\bar{h}_T f(\mathbf{w}^*, \mathbf{x}) < 0|S)$$

Settings and Notations

The idea is to look at the "error-rate" with respect to $\mathbf{w}^*$:

$$\epsilon_t = \Pr(h_t(\mathbf{x}) f(\mathbf{w}^*, \mathbf{x}) < 0)$$
$$\bar{\epsilon} = \Pr(\bar{h}(\mathbf{x}) f(\mathbf{w}^*, \mathbf{x}) < 0)$$
$$\bar{\epsilon}_T = \Pr(\bar{h}_T f(\mathbf{w}^*, \mathbf{x}) < 0 | S)$$

Also, it is easy to see:

$$\bar{\epsilon} = \frac{\sum_{t=1}^{k} \epsilon_t}{k}, \quad \bar{\epsilon}_T = \frac{\sum_{t=1}^{k} \mathbf{1}(t \text{ not pruned}) \epsilon_t}{|\{t : \text{t not pruned}\}|}$$

Settings and Notations

The idea is to look at the "error-rate" with respect to $\mathbf{w}^*$:

$$\epsilon_t = \Pr(h_t(\mathbf{x})f(\mathbf{w}^*, \mathbf{x}) < 0)$$
$$\bar{\epsilon} = \Pr(\bar{h}(\mathbf{x})f(\mathbf{w}^*, \mathbf{x}) < 0)$$
$$\bar{\epsilon}_T = \Pr(\bar{h}_T f(\mathbf{w}^*, \mathbf{x}) < 0|S)$$

Also, it is easy to see:

$$\bar{\epsilon} = \frac{\sum_{t=1}^{k} \epsilon_t}{k}, \quad \bar{\epsilon}_T = \frac{\sum_{t=1}^{k} \mathbf{1}(t \text{ not pruned})\epsilon_t}{|\{t : \text{t not pruned}\}|}$$

So, what is the relationship between $\bar{\epsilon}_T$ and classification error?

Theorem 1

Assuming $p_D(y|\mathbf{x}) \in \{0,1\}$*, it holds for any teacher* t *that*

$$\epsilon_t = \Pr_{(\mathbf{x},y)\sim D}(yf(\mathbf{w}^*,\mathbf{x}) < 0) + \mathbb{E}_{(\mathbf{x},y)}[e_t(\mathbf{x})\mathit{sign}(yf(\mathbf{w}^*,\mathbf{x}))]$$

Algorithms and Theoretical Foundations

Corollary 2

Assume that for any teacher t, $e_t(\mathbf{x}) \equiv e_t$ *is a constant independent of* $\mathbf{x}$. *If* $\Pr(sign(f(\mathbf{w}^*, \mathbf{x})) \neq y) < 1/2$, *then* $\{\epsilon_t\}, \bar{\epsilon}, \bar{\epsilon}_T$ *are equivalent to* $\{e_t\}, \bar{e}, \bar{e}_T$ *respectively, up to a uniform, monotonically increasing linear transformation.*

Algorithms and Theoretical Foundations

Corollary 2

Assume that for any teacher t, $e_t(\mathbf{x}) \equiv e_t$ is a constant independent of $\mathbf{x}$. If $\Pr(sign(f(\mathbf{w}^*, \mathbf{x})) \neq y) < 1/2$, then $\{\epsilon_t\}, \bar{\epsilon}, \bar{\epsilon}_T$ are equivalent to $\{e_t\}, \bar{e}, \bar{e}_T$ respectively, up to a uniform, monotonically increasing linear transformation.

- This means, $\mathbf{w}^*$ does not have to be particularly good, an error-rate smaller than $1/2$ suffices.

Pruning Algorithm

Motivated by Theorem 1, we consider the following simple algorithm to prune teachers:

Pruning Algorithm

Motivated by Theorem 1, we consider the following simple algorithm to prune teachers:

- Train a classifier $\mathbf{w}'$ on the entire dataset and prune away any teacher for which

$$\frac{\sum_{i \in S_t} \mathbf{1}(h_t(\mathbf{x}_i) f(\mathbf{w}', \mathbf{x}_i) < 0)}{|S_t|} > T$$

  for some threshold $T \in (0, 1)$

Pruning Algorithm

Motivated by Theorem 1, we consider the following simple algorithm to prune teachers:

- Train a classifier $\mathbf{w}'$ on the entire dataset and prune away any teacher for which

$$\frac{\sum_{i \in S_t} \mathbf{1}(h_t(\mathbf{x}_i) f(\mathbf{w}', \mathbf{x}_i) < 0)}{|S_t|} > T$$

  for some threshold $T \in (0, 1)$
- Essentially, this calculates a rough empirical estimate of ϵ_t, and removes all teachers where this estimate exceeds the threshold T

Pruning Algorithm

Motivated by Theorem 1, we consider the following simple algorithm to prune teachers:

- Train a classifier $\mathbf{w}'$ on the entire dataset and prune away any teacher for which

$$\frac{\sum_{i \in S_t} \mathbf{1}(h_t(\mathbf{x}_i) f(\mathbf{w}', \mathbf{x}_i) < 0)}{|S_t|} > T$$

  for some threshold $T \in (0, 1)$
- Essentially, this calculates a rough empirical estimate of ϵ_t, and removes all teachers where this estimate exceeds the threshold T
- The question is: Can this actually help???

Pruning Can Help

Theorem 3

Assume we use the pruning procedure described previously. Also, let $F : [0,1] \to [0,1]$ be a cumulative distribution function, such that $F(a) = \frac{1}{k}\sum_{t=1}^{k} \mathbf{1}(\epsilon_t \leq a)$. Let $P \sim F(\cdot)$, and let $N \sim Poi(m/k)$ be a Poisson random variable with parameter m/k. If we assume $m/k = \Theta(1)$ as m, k increase, it holds that

$$\bar{\epsilon} = \mathbb{E}_P[P]$$

and with probability at least $1 - \delta$ over the training sample

$$\bar{\epsilon}_T \leq \frac{\mathbb{E}_{P,N}[\Pr(X_N^P \leq NT)P] + r(m,\delta)}{\mathbb{E}_{P,N}[\Pr(X_N^P \leq NT)] - r(m,\delta)}$$

where X_N^P is a binomial random variable, representing sum of N independent Bernoulli random variables with parameter P, and

$$r(m,\delta) = O\left(\sqrt{\frac{\log(6/\delta)}{m}}\right)$$

Pruning Can Help

Pruning Can't Hurt

Another question is:
Can we guarantee that $\bar{\epsilon}_T$ is never considerably larger than $\bar{\epsilon}$?

Pruning Can't Hurt

Another question is:
Can we guarantee that $\bar{\epsilon}_T$ is never considerably larger than $\bar{\epsilon}$?

Theorem 4

In the setting of Theorem 3, it holds for any $\{\epsilon_t\}$ that

$$\bar{\epsilon}_T \leq \bar{\epsilon} + \frac{2r(m,\delta)}{\mathbb{E}_{P,N}[\Pr(X_N^P \leq NT)] - r(m,\delta)}$$

where

$$r(m,\delta) = O\left(\sqrt{\frac{\log(6/\delta)}{m}}\right)$$

Setting the Threshold T

One final question is, how to choose the threshold T?

Setting the Threshold T

One final question is, how to choose the threshold T?

Corollary 5

In the setting of Theorem 1, a sufficient condition for $e_t > \bar{e}$ is

$$\epsilon_t > \Pr_{(\mathbf{x},y)\sim D}(yf(\mathbf{w}^*,\mathbf{x}) < 0) + \bar{e}$$

Setting the Threshold T

One final question is, how to choose the threshold T?

Corollary 5

In the setting of Theorem 1, a sufficient condition for $e_t > \bar{e}$ *is*

$$\epsilon_t > \Pr_{(\mathbf{x},y)\sim D}(yf(\mathbf{w}^*,\mathbf{x}) < 0) + \bar{e}$$

- This corollary implies that, if ϵ_t is larger than a certain quantity, it is definitely worse than average
- This suggests a reasonable choice for T is:

$$\Pr_{(\mathbf{x},y)\sim D}(yf(\mathbf{w}',\mathbf{x}) < 0) + \bar{e}$$

Reusing the Cleaned Dataset

- If pruning is successful, we expect to have a cleaner dataset

Reusing the Cleaned Dataset

- If pruning is successful, we expect to have a cleaner dataset
- A more accurate classifier is thereby obtainable

Reusing the Cleaned Dataset

- If pruning is successful, we expect to have a cleaner dataset
- A more accurate classifier is thereby obtainable
- However, the pruning is data-dependent, therefore generalization will be an issue!

Reusing the Cleaned Dataset

Fortunately, we can address this easily:

- First, randomly split S into S_1 and S_2

Reusing the Cleaned Dataset

Fortunately, we can address this easily:

- First, randomly split S into S_1 and S_2
- Second, we get low-quality teacher set B_1 and B_2 according to S_1 and S_2

Reusing the Cleaned Dataset

Fortunately, we can address this easily:

- First, randomly split S into S_1 and S_2
- Second, we get low-quality teacher set B_1 and B_2 according to S_1 and S_2
- Third, use B_1 clean S_2 to get S_2' and use B_2 clean S_1 to get S_1'

Reusing the Cleaned Dataset

Fortunately, we can address this easily:

- First, randomly split S into S_1 and S_2
- Second, we get low-quality teacher set B_1 and B_2 according to S_1 and S_2
- Third, use B_1 clean S_2 to get S_2' and use B_2 clean S_1 to get S_1'
- Finally, train a classifier on S', where

$$S' = S_1' \cup S_2'$$

Experiments

Experiments

- The data-pruning approach is tested using *Amazon.com's Mechanical Turk*

Experiments

- The data-pruning approach is tested using *Amazon.com's Mechanical Turk*
- We create an unlabeled set of over $8,000$ examples, each consists
  - A search engine query
  - An Internet URL

Experiments

- The data-pruning approach is tested using *Amazon.com's Mechanical Turk*
- We create an unlabeled set of over $8,000$ examples, each consists
  - A search engine query
  - An Internet URL
- Task was to determine if they are relevant match or not

Experiments

- The data-pruning approach is tested using *Amazon.com's Mechanical Turk*
- We create an unlabeled set of over $8,000$ examples, each consists
  - A search engine query
  - An Internet URL
- Task was to determine if they are relevant match or not
- Each example was labeled by 15 different teachers

Experiments

- The data-pruning approach is tested using *Amazon.com's Mechanical Turk*
- We create an unlabeled set of over $8,000$ examples, each consists
  - A search engine query
  - An Internet URL
- Task was to determine if they are relevant match or not
- Each example was labeled by 15 different teachers
- A total of 375 individual teachers contributed to the dataset

Experiments

μ	$\mu = \infty$(original)	$\mu = 200$	$\mu = 50$
No. of Teachers	375	881	2509
Typical Label / Teacher	NA	14	4

Table: Description of 3 Datasets

- Parameter μ: each teacher labels at most μ examples

Experiments

μ	$\mu = \infty$(original)	$\mu = 200$	$\mu = 50$
No. of Teachers	375	881	2509
Typical Label / Teacher	NA	14	4

Table: Description of 3 Datasets

- Parameter μ: each teacher labels at most μ examples
- The average of 15 labels are trated as ground truth

Experiments

μ	$\mu = \infty$(original)	$\mu = 200$	$\mu = 50$
No. of Teachers	375	881	2509
Typical Label / Teacher	NA	14	4

Table: Description of 3 Datasets

- Parameter μ: each teacher labels at most μ examples
- The average of 15 labels are trated as ground truth
- The training algorithm is well-tuned linear SVM

Experiments

Q & A

Thank you very much!

Any Questions?