Learnability of Linear Thresholds from Label Proportions

Rishi Saket

Google Research
Bangalore, India

NeurIPS 2021

Motivation and Problem Definition

PAC model [Valiant'84] :- Given a training samples ($\mathbf{x}$, f($\mathbf{x}$)) ~ distn. D, efficiently output h s.t. $\Pr_D[h(\mathbf{x}) = f(\mathbf{x})] \geq 1 - \delta$, $\forall\ \delta > 0$. (f, h : $\mathbb{R}^d$, $\{0,1\}^d \rightarrow \{0,1\}$)

Motivation and Problem Definition

PAC model [Valiant'84] :- Given a training samples ($\mathbf{x}$, f($\mathbf{x}$)) ~ distn. D, efficiently output h s.t. $Pr_D[h(\mathbf{x}) = f(\mathbf{x})] \geq 1 - \delta$, $\forall$ $\delta > 0$. (f, h : $\mathbb{R}^d$, $\{0,1\}^d \rightarrow \{0,1\}$)

- If so, class of { f } can be learnt by class of { h }.

Motivation and Problem Definition

PAC model [Valiant'84] :- Given a training samples ($\mathbf{x}$, f($\mathbf{x}$)) ~ distn. D, efficiently output h s.t. $\Pr_D[h(\mathbf{x}) = f(\mathbf{x})] \geq 1 - \delta$, $\forall$ $\delta > 0$. (f, h : $\mathbb{R}^d$, $\{0,1\}^d \rightarrow \{0,1\}$)

- If so, class of { f } can be learnt by class of { h }.
- linear threshold functions (LTFs) a.k.a. *halfspaces* can be learnt by LTFs

LTF : pos($\langle \mathbf{r}, \mathbf{x} \rangle$+ c) where pos(a) = 1 if a > 0, 0 otherwise.

E.g. pos($2x_1 + 3x_2 - x_3 + 2$)

Motivation and Problem Definition

PAC model [Valiant'84] :- Given a training samples ($\mathbf{x}$, f($\mathbf{x}$)) ~ distn. D, efficiently output h s.t. $\Pr_D[h(\mathbf{x}) = f(\mathbf{x})] \geq 1 - \delta$, $\forall\ \delta > 0$. (f, h : $\mathbb{R}^d$, $\{0,1\}^d \rightarrow \{0,1\}$)

- If so, class of { f } can be learnt by class of { h }.
- linear threshold functions (LTFs) a.k.a. *halfspaces* can be learnt by LTFs
- 2-term DNFs can be learnt by degree-2 polynomial threshold fns. (PTFs)

OR of two ANDs

E.g. $(x_1 \wedge \neg x_2) \vee (x_2 \wedge \neg x_3 \wedge x_4)$

Degree-t PTF : pos(p($\mathbf{x}$)) where p($\mathbf{x}$) is deg.-t polynomial

E.g. deg.-2 PTF: pos($x^2_1 + x_2 - 4x_3 + 7$)

Motivation and Problem Definition

PAC model [Valiant'84] :- Given a training samples ($\mathbf{x}$, f($\mathbf{x}$)) ~ distn. D, efficiently output h s.t. $Pr_D[h(\mathbf{x}) = f(\mathbf{x})] \geq 1 - \delta$, $\forall\ \delta > 0$. (f, h : $\mathbb{R}^d$, $\{0,1\}^d \to \{0,1\}$)

- If so, class of { f } can be learnt by class of { h }.

Motivation and Problem Definition

PAC model [Valiant'84] :- Given a training samples ($\mathbf{x}$, f($\mathbf{x}$)) ~ distn. D, efficiently output h s.t. $\Pr_D[h(\mathbf{x}) = f(\mathbf{x})] \geq 1 - \delta$, $\forall\ \delta > 0$. (f, h : $\mathbb{R}^d$, $\{0,1\}^d \rightarrow \{0,1\}$)

- If so, class of { f } can be learnt by class of { h }.

What if only aggregate training labels for collections (*bags*) of feature vecs.?

- Privacy [Wojtusiak et al.'11] [Rueping'10] constraints
- Labeling Cost [Chen et al.'04], lack of instrumentation [Dery et al.'17]

Learning from Label Proportions (LLP)

- Feature-vector space $\mathscr{X} = \mathbb{R}^d, \{0,1\}^d$ Bags $\mathscr{B} = 2^{\mathscr{X}}$.

Learning from Label Proportions (LLP)

- Feature-vector space $\mathscr{X}$ = $\mathbb{R}^d$, $\{0,1\}^d$ Bags $\mathscr{B} = 2^{\mathscr{X}}$.
- $f : \mathscr{X} \rightarrow \{0,1\}$, define $\sigma(B,f) \coloneqq \mathrm{Avg}\{f(\mathbf{x}) : \mathbf{x} \in B\}$ for $B \in \mathscr{B}$

Learning from Label Proportions (LLP)

- Feature-vector space $\mathscr{X} = \mathbb{R}^d, \{0,1\}^d$ Bags $\mathscr{B} = 2^{\mathscr{X}}$.
- $f : \mathscr{X} \to \{0,1\}$, define $\sigma(B,f) \coloneqq \mathrm{Avg}\{f(\mathbf{x}) : \mathbf{x} \in B\}$ for $B \in \mathscr{B}$
- Training examples $(B, \sigma(B,f))$, goal is to train h consistent with f.
- $h : \mathscr{X} \to \{0,1\}$ *satisfies* B if $\sigma(B,h) = \sigma(B,f)$

Learning from Label Proportions (LLP)

- Feature-vector space $\mathscr{X} = \mathbb{R}^d, \{0,1\}^d$ Bags $\mathscr{B} = 2^{\mathscr{X}}$.
- $f : \mathscr{X} \to \{0,1\}$, define $\sigma(B,f) \coloneqq \mathrm{Avg}\{f(\mathbf{x}) : \mathbf{x} \in B\}$ for $B \in \mathscr{B}$
- Training examples $(B, \sigma(B,f))$, goal is to train h consistent with f.
- $h : \mathscr{X} \to \{0,1\}$ *satisfies* B if $\sigma(B,h) = \sigma(B,f)$

Goal: Given $(B_k, \sigma(B_k,f))$ sampled from some distribution, $(k=1,...,m)$

find hypothesis $h : \mathscr{X} \to \{0,1\}$ maximizing # satisfied bags B_k.

Learning from Label Proportions (LLP)

- Feature-vector space $\mathscr{X} = \mathbb{R}^d, \{0,1\}^d$ Bags $\mathscr{B} = 2^{\mathscr{X}}$.
- $f : \mathscr{X} \to \{0,1\}$, define $\sigma(B,f) \coloneqq \mathrm{Avg}\{f(\mathbf{x}) : \mathbf{x} \in B\}$ for $B \in \mathscr{B}$
- Training examples $(B, \sigma(B,f))$, goal is to train h consistent with f.
- $h : \mathscr{X} \to \{0,1\}$ *satisfies* B if $\sigma(B,h) = \sigma(B,f)$

Goal: Given $(B_k, \sigma(B_k,f))$ sampled from some distribution, $(k=1,...,m)$

find hypothesis $h : \mathscr{X} \to \{0,1\}$ maximizing # satisfied bags B_k.

- Weaker notions of bag consistency [Yu et al.'14]
- Strict consistency makes sense for small bags.

Learnability of Linear Thresholds

Our study: f is LTF and h is also LTF

Learnability of Linear Thresholds

Our study: f is LTF and h is also LTF

PAC Learning (bags of size 1 only)

- LTF is efficiently learnable using LTF to arbitrary accuracy
- Linear Programming can find LTF satisfying all training examples.

Learnability of Linear Thresholds

Our study: f is LTF and h is also LTF

PAC Learning (bags of size 1 only)

- LTF is efficiently learnable using LTF to arbitrary accuracy
- Linear Programming can find LTF satisfying all training examples.

Hardness of PAC learning LTFs

- In presence of adversarial ε-noise, NP-hard to compute any constant degree PTF with accuracy $½ + \delta$, for any const. $\delta > 0$ [Bhattacharyya Ghoshal S.'18]
  - Improves on [Guruswami-Raghavendra'06, Feldman et al.'06, Diakonikolas et al.'11]

Learnability of Linear Thresholds

Our study: f is LTF and h is also LTF

PAC Learning (bags of size 1 only)

- LTF is efficiently learnable using LTF to arbitrary accuracy
- Linear Programming can find LTF satisfying all training examples.

Hardness of PAC learning LTFs

- In presence of adversarial ε-noise, NP-hard to compute any constant degree PTF with accuracy ½ + δ, for any const. δ > 0 [Bhattacharyya Ghoshal S.'18]
    - Improves on [Guruswami-Raghavendra'06, Feldman et al.'06, Diakonikolas et al.'11]

Question: In the noiseless LLP setting, what is the complexity of learning LTF?

LLP Learnability of Linear Thresholds (LLP-LTF)

Question: In the noiseless LLP setting, what is the complexity of learning LTF?

Our Answer: Drastically harder, even if only bags of size ≤ 2 are allowed.

LLP Learnability of Linear Thresholds (LLP-LTF)

Question: In the noiseless LLP setting, what is the complexity of learning LTF?

Our Answer: Drastically harder, even if only bags of size ≤ 2 are allowed.

- Linear programming doesn’t work (don’t know feature-vector labels)

LLP Learnability of Linear Thresholds (LLP-LTF)

Question: In the noiseless LLP setting, what is the complexity of learning LTF?

Our Answer: Drastically harder, even if only bags of size ≤ 2 are allowed.

- Linear programming doesn't work (don't know feature-vector labels)

<u>Our Algorithmic Results</u>:

Given instance $(\{(B_k, \sigma(B_k,f))\} : k = 1,...,m)$ s.t. $|B_k| \leq 2$, f is unknown LTF:

- Efficient algorithm that finds an LTF h satisfying ⅖ fraction of all the bags.

LLP Learnability of Linear Thresholds (LLP-LTF)

Question: In the noiseless LLP setting, what is the complexity of learning LTF?

Our Answer: Drastically harder, even if only bags of size ≤ 2 are allowed.

- Linear programming doesn't work (don't know feature-vector labels)

<u>Our Algorithmic Results</u>:

Given instance $(\{(B_k, \sigma(B_k,f))\} : k = 1,...,m)$ s.t. $|B_k| \leq 2$, f is unknown LTF:

- Efficient algorithm that finds an LTF h satisfying ⅖ fraction of all the bags.
- If all bags are *non-monochromatic* then h satisfies ½ frac.
- (Trivial) easy to to find an LTF satisfying all monochromatic bags.

LLP Learnability of Linear Thresholds (LLP-LTF)

Question: In the noiseless LLP setting, what is the complexity of learning LTF?

Our Answer: Drastically harder, even if only bags of size ≤ 2 are allowed.

- Linear programming doesn't work (don't know feature-vector labels)

<u>Our Algorithmic Results</u>:

Given instance $(\{(B_k, \sigma(B_k, f))\} : k = 1,...,m)$ s.t. $|B_k| \leq 2$, f is unknown LTF:

- Efficient algorithm that finds an LTF h satisfying $\frac{2}{5}$ fraction of all the bags.
- If all bags are *non-monochromatic* then h satisfies $\frac{1}{2}$ frac.
- (Trivial) easy to find an LTF satisfying all monochromatic bags.

Question: Can we do better?

LLP Learnability of Linear Thresholds (LLP-LTF)

<u>Our (Main) Hardness Result</u>:

Given an instance of LLP-LTF over $\mathscr{X} = \{0,1\}^d$

- consisting only of non-monochromatic bags,
- each bag of size 2, s.t.
- there is a monotone OR that satisfies all bags,

LLP Learnability of Linear Thresholds (LLP-LTF)

Our (Main) Hardness Result:

Given an instance of LLP-LTF over $\mathscr{X} = \{0,1\}^d$

- consisting only of non-monochromatic bags,
- each bag of size 2, s.t.
- there is a monotone OR that satisfies all bags,

it is NP-hard to find *any* boolean function of q LTFs that

satisfies $(1/2 + \delta)$-fraction of the bags, for any constants $q \in \mathbb{Z}^+$, $\delta > 0$.

LLP-LTF is provably hard to approximate even for a very special case.

LLP Learnability of Linear Thresholds (LLP-LTF)

Proof idea: Start with Label-Cover $\mathscr{L}$: NP-hard 2-variable CSP

Replace each variable of $\mathscr{L}$ with a group of coordinates.

Transform each “edge” of $\mathscr{L}$ into a sub-instance of LLP-LTF.

LLP Learnability of Linear Thresholds (LLP-LTF)

Proof idea: Start with Label-Cover $\mathcal{L}$: NP-hard 2-variable CSP

Replace each variable of $\mathcal{L}$ with a group of coordinates.

Transform each "edge" of $\mathcal{L}$ into a sub-instance of LLP-LTF.

Use a bespoke *dictatorship test* :

- any satisfying labeling to the edge corresponds to solution of sub-instance
- any good enough solution to sub-instance can be *independently decoded* to a satisfying labeling to the edge (with significant probability).

Final instance : union of all sub-instances.

Tools: anti-concentration, multi-dim. Berry-Esseen

Algorithm (Bags of size ≤ 2)

m : # bags, s : # non-monochromatic bags

(Trivial Algorithm) Given instance of LLP-LTF easy to find (using LP) an LTF which satisfies all the (m-s) monochromatic bags

Algorithm (Bags of size ≤ 2)

m : # bags, s : # non-monochromatic bags

(Trivial Algorithm) Given instance of LLP-LTF easy to find (using LP) an LTF which satisfies all the (m-s) monochromatic bags

Main Algorithm $\mathscr{A}$: Given LLP-LTF instance computes in poly-time an LTF that satisfies in expectation (s/2 + (m-s)/4) bags

⅖ -approximation: If (m-s) ≥ (⅖)m use Trivial Algo., else use $\mathscr{A}$.

Main Algorithm $\mathscr{A}$

Append 1 to each feature vector $\mathbf{x}$ so that the satisfying LTF is $\mathrm{pos}(\langle \mathbf{r}^*, \mathbf{x} \rangle)$

Assume that training points are classified with non-zero margin by $\mathrm{pos}(\langle \mathbf{r}^*, \mathbf{x} \rangle)$

Main Algorithm $\mathscr{A}$

Append 1 to each feature vector $\mathbf{x}$ so that the satisfying LTF is $\mathrm{pos}(\langle \mathbf{r}^*, \mathbf{x} \rangle)$

Assume that training points are classified with non-zero margin by $\mathrm{pos}(\langle \mathbf{r}^*, \mathbf{x} \rangle)$

Suppose $\mathbf{x}^i$ and $\mathbf{x}^j$ are in a bag B. Then,

$\langle \mathbf{r}^*, \mathbf{x}^i \rangle \langle \mathbf{r}^*, \mathbf{x}^j \rangle < 0$ if B is non-monochromatic and $\langle \mathbf{r}^*, \mathbf{x}^i \rangle \langle \mathbf{r}^*, \mathbf{x}^j \rangle > 0$ o/w

Main Algorithm $\mathscr{A}$

Append 1 to each feature vector $\mathbf{x}$ so that the satisfying LTF is pos($\langle \mathbf{r}^*, \mathbf{x} \rangle$)

Assume that training points are classified with non-zero margin by pos($\langle \mathbf{r}^*, \mathbf{x} \rangle$)

Suppose $\mathbf{x}^i$ and $\mathbf{x}^j$ are in a bag B. Then,

$\langle \mathbf{r}^*, \mathbf{x}^i \rangle \langle \mathbf{r}^*, \mathbf{x}^j \rangle < 0$ if B is non-monochromatic and $\langle \mathbf{r}^*, \mathbf{x}^i \rangle \langle \mathbf{r}^*, \mathbf{x}^j \rangle > 0$ o/w

i.e. the following SDP over symmetric psd $\mathbf{R}$:

$$(\mathbf{x}^i)^T \mathbf{R}\, \mathbf{x}^j < 0 \text{ for all non-mon. bags } \{\mathbf{x}^i, \mathbf{x}^j\}$$

$$(\mathbf{x}^i)^T \mathbf{R}\, \mathbf{x}^j > 0 \text{ for all mon. bags } \{\mathbf{x}^i, \mathbf{x}^j\}$$

is feasible with at least one solution $\mathbf{R} = \mathbf{r}^* \mathbf{r}^{*T}$

Main Algorithm $\mathscr{A}$

Solve the SDP for symmetric psd $\mathbf{R}$ and factor it as $\mathbf{R} = \mathbf{A}^{\mathsf{T}}\mathbf{A}$

Thus,

$$\langle \mathbf{A}\mathbf{x}^i, \mathbf{A}\mathbf{x}^j \rangle < 0 \text{ for all non-mon. bags } \{\mathbf{x}^i, \mathbf{x}^j\}$$

$$\langle \mathbf{A}\mathbf{x}^i, \mathbf{A}\mathbf{x}^j \rangle > 0 \text{ for all mon. bags } \{\mathbf{x}^i, \mathbf{x}^j\}$$

Main Algorithm $\mathscr{A}$

Solve the SDP for symmetric psd $\mathbf{R}$ and factor it as $\mathbf{R} = \mathbf{A}^{\mathsf{T}}\mathbf{A}$

Thus,

$$\langle \mathbf{Ax}^i, \mathbf{Ax}^j \rangle < 0 \text{ for all non-mon. bags } \{\mathbf{x}^i, \mathbf{x}^j\}$$

$$\langle \mathbf{Ax}^i, \mathbf{Ax}^j \rangle > 0 \text{ for all mon. bags } \{\mathbf{x}^i, \mathbf{x}^j\}$$

Sample $\mathbf{g}$ as a random standard Gaussian vector.

$$\Pr_{\mathbf{g}}[\mathrm{pos}(\langle \mathbf{Ax}^i, \mathbf{g} \rangle) \neq \mathrm{pos}(\langle \mathbf{Ax}^j, \mathbf{g} \rangle)] > ½ \text{ for all non-mon. bags } \{\mathbf{x}^i, \mathbf{x}^j\}$$

$$\Pr_{\mathbf{g}}[\mathrm{pos}(\langle \mathbf{Ax}^i, \mathbf{g} \rangle) = \mathrm{pos}(\langle \mathbf{Ax}^j, \mathbf{g} \rangle)] > ½ \text{ for all mon. bags } \{\mathbf{x}^i, \mathbf{x}^j\}$$

Main Algorithm $\mathscr{A}$

Define LTFs $h(\mathbf{x}) = \text{pos}(\langle \mathbf{Ax}, \mathbf{g} \rangle)$, $h'(\mathbf{x}) = \text{pos}(-\langle \mathbf{Ax}, \mathbf{g} \rangle)$

h and h' both satisfy the special non-monochromatic bags.

Main Algorithm $\mathscr{A}$

Define LTFs $h(\mathbf{x}) = \mathrm{pos}(\langle \mathbf{Ax}, \mathbf{g} \rangle)$, $h'(\mathbf{x}) = \mathrm{pos}(-\langle \mathbf{Ax}, \mathbf{g} \rangle)$

h and h' both satisfy the special non-monochromatic bags.

One of h and h' satisfies at least ½ of the special monochromatic bags.

Taking the best out of h and h' gives the desired random LTF.

Future Directions

Bridge the gap b/w $\frac{2}{5}$ (algo) and $\frac{1}{2}$ (hardness) for LLP-LTF on size ≤ 2 bags.

Extend to algo to larger sized bags (possibly more sophisticated techniques).

Other classifiers: degree-d PTFs, DNF formulas, decision trees, neural-nets ..

Thank You!