Neural Active Learning with Performance Guarantee[1]

Presenter: Yikun Ban

ILLINOIS

[1] Wang, Zhilei, et al. "Neural Active Learning with Performance Guarantees." NeurIPS(2021).

Roadmap

- **Background and Problem Definition**
- **Algorithm 1 and Analysis**
- **Algorithm 2**
- **Summary**

Background: Active Learning

- **Supervised learning is at core of deep learning**
- **Labeled data collected via:**
  - Human experts
  - Crowdsourcing
  - Expensive experiments
- **Requirement of labeled data increase, as complexity of model grows**
- **Active learning: Reduce data requirement**

Background: Active Learning

- **Pool-based active learning:**
  - Access to a large set of unlabeled data, while only asking for the labels of a **subset**
- **Streaming-based active learning:**
  - In each round, a data point is given, learner must decide whether to query the label
- **Data assumption: i.i.d. setting**
  - D *over* $X \times Y$
- **Target function assumption:**
  - Parametric setting: Class of functions H with finite VC-dimension, rate of $O(\frac{\nu}{\sqrt{N}})$
  - Non-parameter setting: Wider class of function
    - Existing work suffer from input dimensionality, rate of $O(\frac{d}{\sqrt{N}})$
    - No provable guarantee for deep neural networks (DNNs)

Stream Setting

- Unknown distribution D over X (input space) $\times$ Y (output space)
  - $Y \to [1, -1]$, Binary Classification
  - $X \in R^d$

- Given a sample x $\sim D_X$ and a hypothesis f, define the conditional population loss as:

$$L(f|x) = E_{y \sim D_{Y|x}}[l(f(x), y)|x]$$

where l is the 0-1 loss: $l(f(x), y) = 1\{f(x) \neq y\} \in \{0, 1\}$

- In round $t \in [T] = \{1, \ldots, T\}$, given a sample $x_t \sim D_X$, make a prediction a_t $(f(x_t))$ and decide whether to query x_t for label

Stream to Bandit Setting

- Consider the doubled space $X^2 \in R^{d\times 2}$. Given x_t, transform $x_t \in R^d$ into 2 arms:

$$x_{t,1} = (x_t, 0) \quad and \quad x_{t,-1} = (0, x_t) \qquad \in R^{d\times 2}$$

$x_{t,1}$ is for the first class and $x_{t,-1}$ is for the other class.

- Suppose there exists an unknown function h:

$$P(y_t = 1|x_t) = h(x_{t,1}) \quad and \quad P(y = -1|x_t) = h(x_{t,-1})$$

Stream to Bandit Setting

- let a be the predicted class ($= f(x_t)$), and then we have

$$\mathbb{E}[\ell(a, y_t) \mid x_t] = 1 - h(x_{t,a})$$

where l is the 0-1 loss: $\ell(a, y) = \mathbb{1}\{a \neq y\} \in \{0, 1\}$

- let a be the predicted class ($= f(x)$), and then we have the pseudo regret:

$$R_T = \sum_{t=1}^{T} \Big(\mathbb{E}_t[\ell(a_t, y_t) \mid x_t] - \mathbb{E}[\ell(a_t^*, y_t) \mid x_t] \Big) = \sum_{t=1}^{T} \big(h(x_{t,a_t^*}) - h(x_{t,a_t}) \big) \ ,$$

where a_t is the predicted class and $a_t^* = \arg\max_{a \in \mathcal{Y}} h(x_{t,a})$

- At the same time, minimize the number of labels N_T

Roadmap

- **Background and Problem Definition**
- **Algorithm 1 and Analysis**
- **Algorithm 2**
- **Summary**

Algorithm 1: Frozen NTK

- Neural network model:

$$f(x,\theta) = \sqrt{m} W_n \sigma(\ldots \sigma(W_1 x)) \ ,$$

- Gradient mapping:

$$\phi(x) = g(x;\theta_0)/\sqrt{m},$$

for $t = 1, 2, \ldots, T$

Observe instance $x_t \in \mathcal{X}$ and build $x_{t,a} \in \mathcal{X}^2$, for $a \in \mathcal{Y}$

Set $\mathcal{C}_{t-1} = \{\theta : \|\theta - \theta_{t-1}\|_{Z_{t-1}} \leq \frac{\gamma_{t-1}}{\sqrt{m}}\}$, with $\gamma_{t-1} = \sqrt{\log\det Z_{t-1} + 2\log(1/\delta)} + S$

Set

$$U_{t,a} = \sqrt{m} \max_{\theta \in \mathcal{C}_{t-1}} \langle \phi(x_{t,a}), \theta - \theta_0 \rangle = \boxed{\sqrt{m}\langle \phi(x_{t,a}), \theta_{t-1} - \theta_0 \rangle} + \boxed{\gamma_{t-1} \|\phi(x_{t,a})\|_{Z_{t-1}^{-1}}}$$

Probability Estimation **UCB**

Algorithm 1: Frozen NTK

Uncertainty estimation

Predict $a_t = \arg\max_{a \in \mathcal{Y}} U_{t,a}$

Set $I_t = \mathbb{1}\{|U_{t,a_t} - 1/2| \leq B_t\} \in \{0,1\}$ with $B_t = B_t(S) = 2\gamma_{t-1}\|\phi(x_{t,a_t})\|_{Z_{t-1}^{-1}}$

if $I_t = 1$

Query $y_t \in \mathcal{Y}$, and set loss $\ell_t = \ell(a_t, y_t)$

Update

$$Z_t = Z_{t-1} + \phi(x_{t,a_t})\phi(x_{t,a_t})^\top$$
$$b_t = b_{t-1} + (1 - \ell_t)\phi(x_{t,a_t})$$
$$\theta_t = Z_t^{-1} b_t / \sqrt{m} + \theta_0$$

Ridge Regression

else $Z_t = Z_{t-1},\ b_t = b_{t-1},\ \theta_t = \theta_{t-1},\ \gamma_t = \gamma_{t-1},\ \mathcal{C}_t = \mathcal{C}_{t-1}$.

Analysis: Algorithm 1

- Low-noise condition with exponent $\alpha \geq 0$, constant $c > 0$:

$$\mathbb{P}\left(\left|h((x,0)) - \tfrac{1}{2}\right| < \epsilon\right) \leq c\,\epsilon^{\alpha} \qquad \epsilon \in [0, 0.5]$$

- when $\alpha \to \infty,\ c = 1$: Hard margin condition

$$\mathbb{P}\left(\left|h((x,0)) - \tfrac{1}{2}\right| < \epsilon\right) = 0$$

- when $\alpha = 0, c = 1$: No assumption to D

Regret Analysis

Theorem 1. *Let Algorithm 1 be run with parameters δ, S, m, and n on an i.i.d. sample $(x_1, y_1), \ldots, (x_T, y_T) \sim \mathcal{D}$, where the marginal distribution $\mathcal{D}_{\mathcal{X}}$ fulfills the low-noise condition with exponent $\alpha \geq 0$ w.r.t. a function h that satisfies (1) and such that $\sqrt{2} S_{T,n}(h) \leq S$. Then with probability at least $1 - \delta$ the cumulative regret R_T and the total number of queries N_T are simultaneously upper bounded as follows:*

$$R_T = O\left(L_H^{\frac{\alpha+1}{\alpha+2}} \left(L_H + \log(\log T/\delta) + S^2 \right)^{\frac{\alpha+1}{\alpha+2}} T^{\frac{1}{\alpha+2}} \right)$$

$$N_T = O\left(L_H^{\frac{\alpha}{\alpha+2}} \left(L_H + \log(\log T/\delta) + S^2 \right)^{\frac{\alpha}{\alpha+2}} T^{\frac{2}{\alpha+2}} \right),$$

where $L_H = \log\det(I + H)$, H being the NTK matrix of depth n over the set of points $\{x_{t,a}\}_{t=1,\ldots,T,\, a=\pm 1}$. $\sqrt{m}\,\|\theta^* - \theta_0\|_2 \leq \sqrt{2} S_{T,n}(h)$

- when $\alpha = 0$, i.e., no assumption to D:

$$N_T = O(T) \qquad R_T = O((L_H + \sqrt{L_H} S)\sqrt{T}) \qquad O\left(\tilde{d} \log(T) \sqrt{T}\right)$$

$L_H = \tilde{d} \log(1 + T)$, where $\tilde{d}$ is the effective dimension $\tilde{d} = \dfrac{\log\det(I + H)}{\log(1 + T)} \leq O(m)$

Regret Analysis

Theorem 1. *Let Algorithm 1 be run with parameters δ, S, m, and n on an i.i.d. sample $(x_1, y_1), \ldots, (x_T, y_T) \sim \mathcal{D}$, where the marginal distribution $\mathcal{D}_X$ fulfills the low-noise condition with exponent $\alpha \geq 0$ w.r.t. a function h that satisfies (1) and such that $\sqrt{2}S_{T,n}(h) \leq S$. Then with probability at least $1 - \delta$ the cumulative regret R_T and the total number of queries N_T are simultaneously upper bounded as follows:*

$$R_T = O\left(L_H^{\frac{\alpha+1}{\alpha+2}} \left(L_H + \log(\log T/\delta) + S^2 \right)^{\frac{\alpha+1}{\alpha+2}} T^{\frac{1}{\alpha+2}} \right)$$

$$N_T = O\left(L_H^{\frac{\alpha}{\alpha+2}} \left(L_H + \log(\log T/\delta) + S^2 \right)^{\frac{\alpha}{\alpha+2}} T^{\frac{2}{\alpha+2}} \right),$$

where $L_H = \log\det(I + H)$, H being the NTK matrix of depth n over the set of points $\{x_{t,a}\}_{t=1,\ldots,T,\, a=\pm 1}$.

- when $\alpha \to \infty$, i.e., Hard margin condition

$$R_T = N_T = O(L_H(L_H + S^2))$$

$$L_H = \tilde{d}\log(1+T)$$

Proof Workflow

Lemma 1. *There exists a positive constant C such that for any $\delta \in (0,1)$, if*

$$m \geq CT^4 n^6 \log(2Tn/\delta)/\lambda_0^4$$

then with probability at least $1-\delta$ over the random initialization θ_0, there exists $\theta^* \in \mathbb{R}^p$ for which

$$h(x_{t,a}) = \langle g(x_{t,a};\theta_0), \theta^* - \theta_0 \rangle \quad \text{and} \quad \sqrt{m}\,\|\theta^* - \theta_0\|_2 \leq \sqrt{2} S_{T,n}(h) \tag{4}$$

for all $t \in [T]$, $a \in \mathcal{Y}$, and h.

Lemma 2. *There exists a positive constant C such that for any $\delta \in (0,1)$, if*

$$m \geq CT^6 n^6 \log(Tn/\delta)$$

then with probability at least $1-\delta$ over the random initialization θ_0 we have

$$Z_t = Z_{t-1} + \phi(x_{t,a_t})\phi(x_{t,a_t})^\top \qquad \log\det Z_T \leq \log\det(I+H) + 1\,. \tag{5}$$

Proof Workflow

Lemma 3. *Let the input parameter S in Algorithm 1 be such that $\sqrt{2}S_{T,n}(h) \leq S$, then under event $\mathcal{E}_0$ for any $\delta > 0$, with probability at least $1 - \delta$ over the random noises we have*

$$\|\theta^* - \theta_t\|_{Z_t} \leq \gamma_t / \sqrt{m}$$

for all $t \geq 0$ simultaneously, i.e., $\theta^* \in \mathcal{C}_t$ with high probability simultaneously for all $t \geq 0$.

$$B_t = 2\gamma_{t-1}\|\phi(x_{t,a_t})\|_{Z_{t-1}^{-1}} .$$

Lemma 5. *For any $b > 0$ we have*

$$\sum_{t=1}^{T} b \wedge I_t B_t^2 \leq 8 \left(\log\det Z_T + 2\log(1/\delta) + S^2 + \frac{b}{8} \right) \log\det Z_T .$$

Proof Workflow

$$\widehat{\Delta}_t = U_{t,a_t} - 1/2\,, \qquad \Delta_t = h(x_{t,a_t}) - 1/2\,, \qquad T_\epsilon = \sum_{t=1}^{T} \mathbb{1}\{\Delta_t^2 \le \epsilon^2\}\,,$$

$$\begin{aligned}
R_T &= \sum_{t=1}^{T} I_t\big(h(x_{t,a_t^*}) - h(x_{t,a_t})\big) \\
&= \sum_{t=1}^{T} I_t\big(h(x_{t,a_t^*}) - h(x_{t,a_t})\big)\,\mathbb{1}\{a_t \neq a_t^*\} \\
&\le \sum_{t=1}^{T} I_t\big|h(x_{t,1}) - h(x_{t,-1})\big|\,\mathbb{1}\{a_t \neq a_t^*\} \\
&= 2\sum_{t=1}^{T} I_t|\Delta_t| \\
&= 2\sum_{t=1}^{T} I_t|\Delta_t|\,\mathbb{1}\{|\Delta_t| > \epsilon\} + \boxed{2\sum_{t=1}^{T} I_t|\Delta_t|\,\mathbb{1}\{|\Delta_t| \le \epsilon\}}\,.
\end{aligned}$$

$$2\epsilon T_\epsilon.$$

$$T_\epsilon \le 3T\epsilon^\alpha + O\left(\log\frac{\log T}{\delta}\right)\,,$$

Proof Workflow

$$\widehat{\Delta}_t = U_{t,a_t} - 1/2\,, \qquad \Delta_t = h(x_{t,a_t}) - 1/2\,, \qquad T_\epsilon = \sum_{t=1}^{T} \mathbb{1}\{\Delta_t^2 \le \epsilon^2\}\,,$$

$$\begin{aligned}
2\sum_{t=1}^{T} I_t |\Delta_t| \mathbb{1}\{|\Delta_t| > \epsilon\} &\le \frac{2}{\epsilon}\sum_{t=1}^{T} I_t \Delta_t^2 \wedge \epsilon \qquad 1\{a > b\} \le \frac{a}{b} \text{ and whole term} \le 1 \\
&\le \frac{2}{\epsilon}\sum_{t=1}^{T} I_t B_t^2 \wedge \frac{1}{2} \\
&\le \frac{16}{\epsilon}\left(\log\det Z_T + 2\log(1/\delta) + S^2 + \frac{1}{16}\right)\log\det Z_T \\
&= O\left(\frac{1}{\epsilon}\left(\log\det(I+H) + \log(1/\delta) + S^2\right)\log\det(I+H)\right)\,.
\end{aligned}$$

Lemma 9. *Under event $\mathcal{E}$, for any $\epsilon \in (0, 1/2)$ we have*

$$\begin{aligned}
R_T &\le 2\epsilon T_\epsilon + \frac{16}{\epsilon}\left(\log\det Z_T + 2\log(1/\delta) + S^2 + \frac{1}{16}\right)\log\det Z_T \\
&= O\left(\epsilon T_\epsilon + \frac{1}{\epsilon}\left(\log\det(I+H) + \log(1/\delta) + S^2\right)\log\det(I+H)\right)\,.
\end{aligned}$$

Roadmap

- **Background and Problem Definition**
- **Algorithm 1 and Analysis**
- **Algorithm 2**
- **Summary**

Algorithm 2: Committee of Algorithm 1

$$L_H(L_H + \log(\log T/\delta) + S^2_{T,n}(h)),$$

d S

- Pool of base learners of Algorithm 1 with a pair of parameters (S_i, d_i).

$$\text{the pool } \mathcal{M}_1 = \{(S_{i_1}, d_{i_2})\},$$

$$S_{i_1} = 2^{i_1},\ i_1 = 0, 1, \ldots, O(\log T)$$

$$d_{i_2} = 2^{i_2},\ i_2 = 0, 1, \ldots, O(\log T + \log\log(M \log T/\delta))$$

Algorithm 2: Workflow

for $t = 1, 2, \ldots, T$

Observe instance $x_t \in \mathcal{X}$ and build $x_{t,a} \in \mathcal{X}^2$, for $a \in \mathcal{Y}$

for $i \in \mathcal{M}_t$

Set $I_{t,i} \in \{0, 1\}$ as the indicator of whether base learner i *would* ask for label on x_t

Set $a_{t,i} \in \mathcal{Y}$ as the prediction of base learner i on x_t

Let $B_{t,i} = B_{t,i}(S_i)$ denote the query threshold of base learner i (from Algorithm 1)

Select base learner $i_t \sim \boldsymbol{p}_t = (p_{t,1}, p_{t,2}, \ldots, p_{t,|\mathcal{M}_t|})$, where

$$p_{t,i} = \begin{cases} \dfrac{d_i^{-(\gamma+1)}}{\sum_{j \in \mathcal{M}_t} d_j^{-(\gamma+1)}}, & \text{if } i \in \mathcal{M}_t \\ 0, & \text{otherwise} \end{cases} \qquad \gamma \geq 0$$

Predict $a_t = a_{t,i_t}$

Predict $a_t = a_{t,i_t}$

if $I_{t,i_t} = 1$

Query label $y_t \in \mathcal{Y}$ and send (x_t, y_t) to base learner i_t

$\mathcal{M}_{t+1} = \mathcal{M}_t$

Algorithm 2: Elimination Test 1-2

- Delete these base learners who suffer regret in non-queried rounds with small S

Set $\mathcal{N}_t = \{i \in \mathcal{M}_t : I_{t,i} = 0\}$ `// (1) Disagreement test`
for *all pairs of base learners* $i, j \in \mathcal{N}_t$ *that disagree in their prediction* $(a_{t,i} \neq a_{t,j})$
  Eliminate all learners with smaller S: $\mathcal{M}_{t+1} = \{m \in \mathcal{M}_{t+1} : S_m > \min\{S_i, S_j\}\}$

- Delete these base learners who would not have required query in queried rounds and the regret of whom exceed the small limit.

for *all pairs of base learners* $i, j \in \mathcal{M}_t$ `// (2) Observed regret test`
  Consider rounds where the chosen learner i requested the label but j did not, and i and j disagree in their prediction:

$$\mathcal{V}_{t,i,j} = \{k \in [t] : i_k = i, I_{k,i} = 1, I_{k,j} = 0, a_{k,i} \neq a_{k,j}\}$$

  if $\sum_{k \in \mathcal{V}_{t,i,j}} (\mathbb{1}\{a_{k,i} \neq y_k\} - \mathbb{1}\{a_{k,j} \neq y_k\}) > \sum_{k \in \mathcal{V}_{t,i,j}} (1 \wedge B_{k,i}) + 1.45\sqrt{|\mathcal{V}_{t,i,j}| L(|\mathcal{V}_{t,i,j}|, \delta)}$
    Eliminate base learner i: $\mathcal{M}_{t+1} = \mathcal{M}_{t+1} \setminus \{i\}$

Algorithm 2: Elimination Test 3-4

- Delete these base learners who have queried too many times.

for $i \in \mathcal{M}_t$ // (3) Label complexity test

  Consider rounds where base learner i was played: $\mathcal{T}_{t,i} = \{k \in [t] : i_k = i\}$

  if

$$\sum_{k \in \mathcal{T}_{t,i}} I_{k,i} > \inf_{\epsilon \in (0,1/2]} \left(3\epsilon^{\gamma} |\mathcal{T}_{t,i}| + \frac{1}{\epsilon^2} \sum_{k \in \mathcal{T}_{t,i}} I_{k,i} B_{k,i}^2 \wedge \frac{1}{4} \right) + 2L(|\mathcal{T}_{t,i}|, \delta/(M \log_2(12t)))$$

    Eliminate base learner i: $\mathcal{M}_{t+1} = \mathcal{M}_{t+1} \setminus \{i\}$

- Delete these base learners with loose upper bounds.

for $i \in \mathcal{M}_t$ // (4) d_i test

  if $\sum_{k \in \mathcal{T}_{t,i}} (\frac{1}{2} \wedge I_{k,i} B_{k,i}^2) > 8 d_i$

    Eliminate base learner i: $\mathcal{M}_{t+1} = \mathcal{M}_{t+1} \setminus \{i\}$

Algorithm 2: Analysis

Theorem 2. *Let Algorithm 2 be run with parameters δ, $\gamma \leq \alpha$ with a pool of base learners $\mathcal{M}_1$ of size M on an i.i.d. sample $(x_1, y_1), \ldots, (x_T, y_T) \sim \mathcal{D}$, where the marginal distribution $\mathcal{D}_X$ fulfills the low-noise condition with exponent $\alpha \geq 0$ w.r.t. a function h that satisfies (1) and complexity $S_{T,n}(h)$. Let also $\mathcal{M}_1$ contain at least one base learner i such that $\sqrt{2}S_{T,n}(h) \leq S_i \leq 2\sqrt{2}S_{T,n}(h)$ and $d_i = \Theta(L_H(L_H + \log(M \log T/\delta) + S^2_{T,n}(h)))$, where $L_H = \log\det(I+H)$, being H the NTK matrix of depth n over the set of points $\{x_{t,a}\}_{t=1,\ldots,T,\, a=\pm 1}$. Then with probability at least $1-\delta$ the cumulative regret R_T and the total number of queries N_T are simultaneously upper bounded as follows:*

$$R_T = O\left(M\left(L_H\left(L_H + \log(M \log T/\delta) + S^2_{T,n}(h)\right)\right)^{\gamma+1} T^{\frac{1}{\gamma+2}} + M\, L(T,\delta)\right)$$

$$N_T = O\left(M\left(L_H\left(L_H + \log(M \log T/\delta) + S^2_{T,n}(h)\right)\right)^{\frac{\gamma}{\gamma+2}} T^{\frac{2}{\gamma+2}} + M\, L(T,\delta)\right),$$

where $L(T,\delta)$ is the logarithmic term defined at the beginning of Algorithm 2's pseudocode.

- Reduce S to $S_{T,n}(h)$
- Under control of γ

Roadmap

- **Background and Problem Definition**
- **Algorithm 1 and Analysis**
- **Algorithm 2**
- **Summary**

Summary

❑ Main contributions of this work:

- Transform streaming-based active learn in to 2-arm bandits
- Apply NeuralUCB to this framework, while adding a component to decide whether to query
- Provide an extended algorithm to loose the assumption for S
- Provide performance guarantee

❑ Advantages of bandit-based approaches in active learning:

- Explicitly incorporate exploration components
- Provide rigorous performance analysis

Possible Future Objectives

- **Extend binary classification into K-classification setting: In the scenarios where active learning can apply, bandit can apply**
  - Design the component to decide whether to query
- **Extend streaming setting to pool-based setting:**
  - Given a set of samples, only can query a subset
- **Extend partial feedback to full feedback:**
  - In many cases, we know the feedback of all the arms, while NeuralAC only leverage the feedback of one arm
- **Apply other exploration strategy to active learning, such as NeuralTS, EE-Net**
- **Remove the dependency of NTK and S; Extend to more complicated loss function, such as MSE, CrossEntropy**

Thanks