Mathematical Proofs for SynGuar

July 16, 2021

1 Preliminaries

Problem Setup. We are given an oracle to sample i.i.d. I/O examples from some unknown distribution D, a synthesizer with bounded hypothesis space and the ability to soundly upper bound number of programs consistent with I/O examples, and user-specified (ϵ, δ) parameters that capture the desired generalization guarantee. The synthesis algorithm queries the oracle for as many I/O examples as it needs and terminates with either $None$ or a synthesized function f. We assume that f will satisfy all given I/O examples. We use S to represent all the I/O examples queried by the synthesis algorithm until it terminates. The probability that the synthesizer returns a function f that might not generalize should be under the given small δ, and the randomness is on I/O examples.

Formally, the goal is for a synthesis algorithm to achieve a PAC-style (ϵ, δ) generalization guarantee. We define (ϵ, δ)-synthesizer as the following:

Definition 1 ((ϵ, δ)-synthesizer)**.** *A synthesis algorithm $\mathcal{A}$ with hypothesis space H is an (ϵ, δ)-synthesizer with respect to a target class of functions $\mathcal{C}$ iff for any input distributions D, for all $t \in \mathcal{C}$, $\epsilon \in (0, 0.5)$, $\delta \in (0, 0.5)$, if $\mathcal{A}(S)$ outputs a program $f \in H$ on a set of samples S drawn i.i.d from the D, then:*

$$Pr[\mathcal{A}(S) \text{ outputs } f \text{ such that } error(f) > \epsilon] < \delta$$

A starting point. The number of examples provably sufficient to achieve the (ϵ, δ)-generalization is given by Blumer et al. [1]. We restate this result, which computes sample complexity as a function of (ϵ, δ) and the capacity (or size) of any given hypothesis space H.

Theorem 1.1 (Sample Complexity for (ϵ, δ)-synthesis)**.** *For all $\epsilon \in (0, \frac{1}{2})$, $\delta \in (0, \frac{1}{2})$, hypothesis space H and any target program t, a synthesis algorithm $\mathcal{A}(S)$ which outputs functions consistent with n i.i.d samples is an (ϵ, δ)-synthesizer, if*

$$n > \frac{1}{\epsilon}(\ln |H| + \ln \frac{1}{\delta})$$

Proof. Assume the true concept (may or may not in the hypothesis space) is t and all I/O examples are consistent with t. Now consider a single hypothesis $f \in H$ first. Let $error(f) = L_{(\mathcal{D},t)}(f)$ be the true error (expectation of 0-1 loss) $L_{(\mathcal{D},t)}(f) = \mathbb{E}_{x\sim\mathcal{D}}\,\mathbb{I}[t(x) \neq f(x)]$, and let $L_{(\mathcal{S},t)}(f)$ be the empirical error (empirical average of 0-1 loss) $L_{(\mathcal{S},t)}(f) = \frac{1}{|S|}\sum_{x\in S}\mathbb{I}[t(x) \neq f(x)]$, The following holds:

$$\Pr_{x\in\mathcal{D}}[\mathbb{I}[t(x) \neq f(x)] = 0 \mid L_{(\mathcal{D},t)}(f) \geq \epsilon] \leq (1-\epsilon)$$

Now consider a sample S with size n, the following holds:

$$\Pr_{S\in\mathcal{D}^n}[L_{(\mathcal{S},t)}(f) = 0 \mid L_{(\mathcal{D},t)}(f) \geq \epsilon] \leq (1-\epsilon)^n$$

Algorithm 1 SynGuar Synthesis returns a program with error smaller than ϵ with probability higher than $1 - \delta$

1: **procedure** SynGuar(ϵ, δ)
2: $\quad k = 1$ // tunable parameter
3: $\quad g \leftarrow$ PickStoppingCond
4: $\quad S' \leftarrow \varnothing, s \leftarrow 0$
5: $\quad size_H \leftarrow$ ComputeSize(H)
6: $\quad n \leftarrow g(size_H)$
7: $\quad$ **while** $s \leq n$ **do**
8: $\quad\quad S' \leftarrow S' \cup$ sample(k)
9: $\quad\quad H_{S'} \leftarrow$ UpdateHypothesis(S')
10: $\quad\quad size_{H_{S'}} \leftarrow$ ComputeSize($H_{S'}$)
11: $\quad\quad s \leftarrow s + k$
12: $\quad\quad n \leftarrow min(n, s + g(size_{H_{S'}}))$
13: $\quad$ **end while**
14: $\quad m_{H_{S'}} = \frac{1}{\epsilon}(\ln size_{H_{S'}} + ln\frac{1}{\delta})$
15: $\quad T \leftarrow$ sample($m_{H_{S'}}(\epsilon, \delta)$)
16: $\quad S \leftarrow S' \cup T$
17: $\quad$ **return** f program in H_S
18: **end procedure**

Then take union bound on all hypothesis in H:

$$\Pr_{S \in \mathcal{D}^n}[\exists f \in H,\ L_{(\mathcal{S},t)}(f) = 0 \ \wedge\ L_{(\mathcal{D},t)}(f) \geq \epsilon] \leq |H|(1-\epsilon)^n < \delta$$
$$\Rightarrow n > \frac{1}{\epsilon}(\ln |H| + \ln \frac{1}{\delta}) \text{ suffices}$$

So with sample size larger than $\frac{1}{\epsilon}(\ln |H| + \ln \frac{1}{\delta})$, with probability at least $1 - \delta$, all $f \in H$ that have true error larger than ϵ have a non-zero empirical loss and will be ruled out and any hypothesis in H that is still consistent with the sample has true error smaller than ϵ.

□

2 Analysis of SynGuar

SynGuar's design is motivated by being able to give a formal generalization guarantee and a bounded sample complexity. For this purpose, we state and prove the following properties:
(P1: Termination) SynGuar always terminates for a finite $|H|$.
(P2: (ϵ, δ) guarantees) If SynGuar returns an f then f is ϵ-far with probability $< \delta$.
(P3: Sample complexity) SynGuar's sample complexity is always within $2\times$ of the optimal.

Theorem 2.1 (P1)**.** SynGuar *always terminates for a finite* $|H|$.

Proof. It suffices to prove that the sampling phase (lines $7 - 13$) of SynGuar terminates in order to show that SynGuar terminates. In each iteration of the sampling phase, let S_i be the queue storing the user-provided examples after each iteration, z_t be the t^{th} example, $S_{i+1} = S_i \cup \{z_{ik+1}, ... z_{ik+k}\}$ and $S_0 = \varnothing$. For each S_i, H_{S_i} determines the set of consistent hypothesis that satisfy S_i. Let N_i be the limit of the number of I/O examples n for the sampling phase after iteration i. For iterations i and j where $i < j$ and $\forall g : \mathbb{N} \rightarrow \mathbb{Z}$ such that g is monotonically non-decreasing, the following holds:

$$S_i \subset S_j \Rightarrow |H_{S_j}| \leq |H_{S_i}| \Rightarrow g(|H_{S_j}|) \leq g(|H_{S_i}|)$$
$$N_j \leq \min\{N_i, |S_j| + g(|H_{S_j}|)\} \leq N_i \text{ (see line 13 in Alg. 1)}$$

Therefore, if $N_0 \leq g(|H|)$ then the loop will terminate at some iteration p such that $N_p < |S_p| \leq N_p + k \leq N_0 + k$. $\square$

Theorem 2.2 (P2)**.** *If* SynGuar *(ϵ, δ) returns the synthesized program f then f is ϵ-far with probability $< \delta$.*

Proof. By Theorem 2.1, we know that the sampling phase terminates with S' samples (see line 14). In lines $14 - 16$ SynGuar samples an additional number of I/O examples required to generalize and then synthesizes a program after seeing the additional samples. Therefore, Theorem 2.2 follows from Theorem 1.1. $\square$

In order to prove the last property, we define a new quantity $\omega(Q)$. It is the smallest sample size taken by SynGuar (ϵ, δ) for any non-decreasing g used for a sequence of I/O examples Q.

Definition 2 (Smallest dynamic sample size)**.** *For any infinite sampled sequence of examples Q, let* Prefix(Q, g) *be the prefix of Q at which* SynGuar *(ϵ, δ) terminates. Then,*

$$\omega(Q) = \inf\{|m_g| \ : \ \forall g, \ m_g = \textsc{Prefix}(Q, g)\}$$

Theorem 2.3 (P3)**.** SynGuar *uses no more than $2\omega(Q)$ examples on any Q when the result is not None with $g(x) = g_0(x) = \max\{0, \ \frac{1}{\epsilon}(\ln(x) - \ln(\frac{1}{\delta}))\}$ and $k \leq \frac{1}{2\epsilon} \ln \frac{1}{\delta}$.*

Proof. Let $S' \subset Q$ be the samples in sampling phase. Let P be the samples when $\omega(Q) =$ Prefix(Q, g) for some g and let us call this the best stopping point for SynGuar's sampling phase on Q. Then $\omega(Q) = |P| + \frac{1}{\epsilon}(\ln |H_P| + ln\frac{1}{\delta})$. Let $g_0(x) = \max\{0, \ \frac{1}{\epsilon}(\ln(x) - \ln(1/\delta))\}$, $\gamma(Q) =$ Prefix(Q, g_0) and $S' \subset Q$ be the samples in sampling phase.

If P is the samples for the best stopping point, then

$$\omega(Q) = |P| + \frac{1}{\epsilon}(\ln |H_P| + ln\frac{1}{\delta})$$

Case 1: SynGuar using g_0 is stopping earlier in phrase 1 than the best possible stopping point, $|S'| \leq |P|$

$$\begin{aligned}
\gamma(Q) - 2 \cdot \omega(Q) &= |S'| - 2 \cdot |P| + \frac{1}{\epsilon} \cdot (\ln |H_{S'}| - 2 \cdot \ln |H_P|) \\
&\quad - \frac{1}{\epsilon} \cdot \ln \frac{1}{\delta} \\
&\leq -|P| - \frac{1}{\epsilon} \cdot \ln \frac{1}{\delta} + \frac{1}{\epsilon} \cdot (\ln |H_{S'}|) \\
&\text{since } |S'| \leq |P|
\end{aligned}$$

Observe that, for $g_0(x) = \max\{0, \ \frac{1}{\epsilon}(\ln(x) - \ln(1/\delta))\}$ the $|S'| \geq \frac{1}{\epsilon} \cdot (\ln |H_{S'}| - \ln \frac{1}{\delta})$ (see, step 12 in Algorithm 2)

$$\gamma(Q) - 2 \cdot \omega(Q) \leq -|P| + |S'| \leq 0$$

Case 2: SynGuar using g_0 is stopping after the best possible stopping point in phrase 1, $|S'| > |P|$

In this case, $|H_P| \geq |H_{S'}|$. Observe that $|S'| \leq \omega(Q)$. Because for SynGuar using g_0, after seeing P, it will take no more than $g_0(|H_P|) + 2k = \max\{0, \frac{1}{\epsilon}(\ln |H_P| - ln\frac{1}{\delta})\} + 2k \leq \frac{1}{\epsilon}(\ln |H_P| + ln\frac{1}{\delta})$ examples in phase 1.

$$\begin{aligned}
|S'| &\leq |P| + 2k + g_0(|H_P|) \\
\omega(Q) &= |P| + \frac{1}{\epsilon}(\ln |H_P| + ln\frac{1}{\delta}) \\
\implies |S'| &\leq \omega(Q)
\end{aligned}$$

Now,

$$\begin{aligned}
\gamma(Q) - 2 \cdot \omega(Q) &= |S'| + \frac{1}{\epsilon} \cdot (\ln |H_{S'}| + \ln \frac{1}{\delta}) - 2 \cdot \omega(Q) \\
&\leq \frac{1}{\epsilon} \cdot (\ln |H_{S'}| + \ln \frac{1}{\delta}) - \omega(Q) \quad (\text{by } |S'| \leq \omega(Q)) \\
&\leq -|P| + \frac{1}{\epsilon} \cdot (\ln |H_{S'}| - \ln |H_P|) \\
&\leq 0 \quad (\text{since, } |H_P| \geq |H_{S'}|)
\end{aligned}$$

□

References

[1] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.