Language Generation in the Limit

Background

Kleinberg and Mullainathan's paper was presented at 38th Conference on Neural Information Processing Systems (NeurIPS 2024). It has already led to several additioanl studies, which collectively were the basis of recent tutorial at COLT.

The paper

The introduction

- contrasts "language generation in the limit" with "language identification in the limit".
- relates the comparative ease of generation (as opposed to identification) to the success of LLMs
- explains that a key limitation manifests itself in terms of what it calls the *breadth problem.*

Preliminaries

- U is the universe.
- Let C be a countable class of recursive languages $\{L_1, L_2, ...\}$ where each $L_i \subseteq U$.
- Each $L_i \in C$ has infinite cardinality (so no finite languages).
- Let C_n be the first n languages in C.
- K is the target language in the class. S_t is the set of words seen from K in times steps $1 \ldots t$.
    - **Identification**: In each step, the algorithm observes S_t and must output an index i (its guess for the true language K). The algorithm identifies K in the limit if there is some t^* such that for all steps $t \geq t^*$, the algorithm's guess in step t is an an index i for which $L_i = K$.
- **Generation**: In each step, the algorithm observes S_t and must output a string a_t (its guess for an unseen string in K). The algorithm generates from K in the limit if there is some t^* such that for all steps $t \geq t^*$, the algorithm's guess a_t belongs to $K - S_t$.

An Approach to Generation that Doesn't Work

This section revisits the subset problem identified by Angluin 1980 that we studied. Basically the least $L_i \in C$ which is consistent with S_t will not work because it may be the case that $S_t \subseteq K \subseteq L_i$, and you will be generating incorrect strings.

Generation in the Limit via a Function

- This section is the core of the paper. It proves the following.

  > **Theorem**. For every countable collection of languages C, there is a function f_C from finite subsets of U to elements of U, such that for every enumeration of a language $K \in C$, there is a t^* such that for all $t \geq t^*$, we have $f_C(S_t) \in K - S_t$.

- The proof of this claim constructs the function f_C. The idea is at time step t it considers 'smallest' languages in C_t consistent with the S_t. These are called **critical** (see below).
  - Let L_{n_t} be the language with the largest index n_t, if it exists. If L_{n_t} exists, $f_C(S_t)$ is defined to be the lowest-indexed element of $L_{n_t} - S_t$.
  - If it does not exists (so no languages in C_t are consistent) then $f_C(S_t)$ is defined arbitrarily (for example the least element in U).
- A language L_n is **critical** at step t if L_n is consistent with S_t, and for every language $L_i \in C_n$ that is consistent with S_t, we have $L_n \subseteq L_i$. In other words, not only is L_n consistent with S_t it is a subset of every language in C_n that precedes L_n in the indexing of C.
  - **Lemma**: There is a time step t^+ such that for all $t \geq t^+$, K is critical.
  - **Lemma**: Let $i < j$. If L_i and L_j are both critical at step t then $L_j \subseteq L_i$.
- It follows that at time step t there may be finitely many critical languages $L_{n_1}, L_{n_2}, L_{n_3}, \ldots$ with $n_1 < n_2 < n_3 \ldots$, which means that the sequence is nested by inclusion: $L_{n_1} \supseteq L_{n_2} \supseteq L_{n_3}, \ldots$.
  - Eventually K appears in this list.
  - "But we now arrive at the crucial point, which is that beyond some finite index, all the critical languages are subsets of [K], so it is safe to generate from any of them."

Generation in the Limit via an Algorithm

This section essentially shows that at time step t, $f_C(S_t)$ can be effectively computed using enumeration and language membership queries.

Generation for Finite Collections of Languages

Here they consider the case where C is finite.

- The **closure** of S_t, denoted $\langle S_t \rangle$, is the intersection of all languages in C consistent with S_t.
- "If there is a string in $\langle S_t \rangle - S_t$, then it is always safe for the algorithm to generate such a string; by definition, it must be an unseen string from the true language [K]."
- This allows them to prove the following

> **Theorem.** There is an algorithm with the property that for any finite collection of languages C, there is a number $t(C)$, such that for any language K in C, and any sequence S of at least $t(C)$ distinct elements from K, the algorithm can produce an infinite sequence of distinct strings from $K - S$.

Extension: Prompted Generation in the Limit

Here they consider the situation where at each time step t, the algorithm is not only given a valid string w_t from K but also a prompt string p_t. The algorithm must produce a completion string c_t such that the concatenation $p_t c_t$ belongs to $K - S_t$.

Concluding Remarks

They write, "the solutions we develop highlight interesting tensions between the problem of producing *valid* strings that belong to the target language, and the problem of maintaining *breadth* by not restricting to only a small subset of the target language. Our approaches achieve validity through a strategy that implicitly gives up on breadth, and it is interesting to ask if this is essentially necessary for any method that achieves language generation in the limit."