Generation through the lens of learning theory

Vinod Raman

Joint work with Jiaxun Li and Ambuj Tewari

A new paradigm in machine learning

- For more than 50 years, predictive ML has been a cornerstone for practitioners and theorists
- Tasks like classification and regression have been extensively studied due to applications to:
  - Face recognition
  - Autonomous vehicles
  - Recommendation systems
  - Spam filtering
- Recently, however, a new paradigm in ML has emerged:

Generation.

Generation

- In Generative ML the goal is not to predict but to create.
  - In language modeling, we generate coherent text in response to a prompt
  - In drug development, we create new candidate molecules
  - In movie production, we render new animations
- Generative ML is revolutionizing how we think and do:
  - Natural Language Processing
  - Computer Vision
  - Chemistry/Biology
  - and much more

Great, but where is the (learning) theory?

- The theoretical foundations of Generative ML lag far behind its predictive counterpart.
- One reason is that the generation is fundamentally an unsupervised task.
- This makes it challenging to define a loss function – the primary workhorse of predictive ML.

Our contributions

We aim to close this gap between learning theory and generative machine learning.

1. We unify existing paradigms of generation through a binary hypothesis $H \subseteq \{0,1\}^X$ defined over a countable abstract instance space X.
2. We formalize new paradigms of generation called "uniform" and "non-uniform" generation and provide their characterizations.
3. We show that (uniform) generation and prediction (i.e. PAC and online learnability) are incomparable – there are classes that are generatable but not predictable and vice versa.

The story so far…

- In 1967, Mark Gold studied the problem of language identification in the limit.
- There is a countable set of strings U and a public family of languages $\mathcal{L} = \{L_1, L_2, \dots\}$ where $L_i \subset U$.
- An adversary secretly picks a language $K \in \mathcal{L}$ and begins enumerating the strings $w_1, w_2, \dots$ in K over rounds $t = 1, 2, \dots$
- After observing w_t in round t, you make a prediction $\hat{L}_t \in \mathcal{L}$.
- You identify K in the limit if there exists a $t^\star \in \mathbb{N}$ such that $\hat{L}_s = K$ for all $s \geq t^\star$.
- $\mathcal{L}$ is identifiable in the limit if you can identify every $K \in \mathcal{L}$.

The story so far...

Gold [1967] show that many natural families of languages are not identifiable in the limit.

Theorem 1 (Gold [1967])
The family of regular languages is not identifiable in the limit.

This theorem is often interpreted as a negative result:

Language Identification is hard.

The story so far…

Dana Angluin, in a series of two works, gives a characterization of which language families are identifiable in the limit.

Theorem 2 (Angluin [1979, 1980])
A language family $\mathcal{L}$ is identifiable in the limit if and only if for every $L \in \mathcal{L}$, there exists a finite $S \subset L$ such that for every $L' \in \mathcal{L}$:

$$S \subset L' \Rightarrow L' \not\subset L$$

Angluin's characterization rules out the vast majority of natural language families.

The story so far…

- The Gold-Angluin model inspired a large amount of discussion, including positive and negative criticisms.
- Some argued that an adversarial Nature is unrealistic and proposed relaxations under which identifiability is possible.
- Regardless, the Gold-Angluin model stands as one of the earliest works in machine learning.

The story so far…

44 years later, Jon Kleinberg and Sendhil Mullainathan revisit the classical setup with a modern twist:

What about generation instead of identification?

Is it easier to eventually generate new strings from the secret language as opposed to identifying it?

The KM Model

- There is a countable set of strings U and a public family of languages $\mathcal{L} = \{L_1, L_2, \dots\}$ where $L_i \subset U$ and $|L_i| = \infty$.
- An adversary secretly picks a language $K \in \mathcal{L}$ and begins enumerating the strings $w_1, w_2, \dots$ in K over rounds $t = 1, 2, \dots$
- After observing w_t in round t, you make a prediction $\widehat{w}_t \in U$.
- You generate from K in the limit if there exists a $t^\star \in \mathbb{N}$ such that $\widehat{w}_s \in K \setminus \{w_1, \dots, w_s\}$ for all $s \geq t^\star$.
- $\mathcal{L}$ is generatable in the limit if you can generate from K for every $K \in \mathcal{L}$.

The KM Model

Remarkably, unlike identification, KM [2024] show that every countable language family is generatable in the limit.

Theorem 3 (KM [2024])
Every countable language family $\mathcal{L}$ is generatable in the limit.

Moreover, for every finite $\mathcal{L}$, generation is possible after observing only a constant number of distinct strings.

Theorem 4 (KM [2024])
If $|\mathcal{L}| < \infty$, there exists a $c \in \mathbb{N}$ such that generation is possible after observing c distinct strings from K.

Beyond language identification/generation

- Both the Gold-Angluin and KM interpret their results with respect to language generation.
- However, nothing is special about languages – the same results hold for generating abstract objects (images, molecules, etc …)
- In statistical learning theory (SLT), we work with abstract spaces.

Can these results be formulated through the lens of SLT?

Generation in the lens of SLT

- Let X be an abstract countable instance space (e.g. images, molecules, …)
- Let $H \subseteq \{0,1\}^X$ be a collection of functions that map instances to a binary label $\{0,1\}$ (e.g. neural networks, transformers, …)
- For every $h \in H$, define $\mathrm{supp}(h) \coloneqq \{x \in X : h(x) = 1\}$.

Generation in the lens of SLT

In Language generation:

- X is the set of valid strings.
- Each $h \in H$ is a language over X parameterized by $\mathrm{supp}(h)$.
- H is a language family.

Assumption 1.
A class $H \subseteq \{0,1\}^X$ satisfies the Uniformly Unbounded Support (UUS) property if $|\mathrm{supp}(h)| = \infty$ for all $h \in H$.

Generatability in the limit

- A generator is a map $G: X^\star \to X$.
- We can use generators to rigorously define what it means for H to be "generatable in the limit."

Definition 1(KM [2024]).
Let $H \subseteq \{0,1\}^X$ be any class that satisfies the UUS property. H is generatable in the limit if there exists a generator G such that for every $h \in H$ and every enumeration $x_1, x_2 \ldots$ of $\mathrm{supp}(h)$ there exists a $t^\star \in \mathbb{N}$ such that for all $s \geq t^\star$

$$G(x_1, \ldots, x_s) \in \mathrm{supp}(h) \setminus \{x_1, \ldots, x_s\}.$$

Beyond “Generatability in the Limit”

- In Definition 1, the time step $t^\star$ after which the Generator must be perfect can depend on:
  1. The hypothesis h chosen by the adversary and
  2. The enumeration of $\mathrm{supp}(h)$.

- This is unsatisfying as, in practice, we would like to know when our generator will be perfect.

- To this end, we can go beyond “generatability in the limit” by swapping the order of quantifiers.

Non-uniform Generatability

Definition 2 (Non-uniform Generatability).
Let $H \subseteq \{0,1\}^X$ be any class that satisfies the UUS property. H is non-uniformly generatable if there exists a generator G such that for every $h \in H$ there exists a $d^\star$ such that for every sequence $x_1, x_2, \ldots \subseteq \mathrm{supp}(h)$, if there exists a $t^\star \in \mathbb{N}$ such that $|\{x_1, \ldots, x_{t^\star}\}| = d^\star$, then for all $s \geq t^\star$

$$G(x_1, \ldots, x_s) \in \mathrm{supp}(h) \setminus \{x_1, \ldots, x_s\}.$$

Uniform Generatability

The strongest form of generatability follows by only allowing a dependence on H.

Definition 3 (Uniform Generatability).
Let $H \subseteq \{0,1\}^X$ be any class that satisfies the UUS property. H is uniformly generatable if there exists a generator G and $d^\star$ such that for every $h \in H$ and every sequence $x_1, x_2 \ldots \subseteq \mathrm{supp}(h)$, if there exists a $t^\star \in \mathbb{N}$ such that $|\{x_1, \ldots, x_{t^\star}\}| = d^\star$, then for all $s \geq t^\star$

$$G(x_1, \ldots, x_s) \in \mathrm{supp}(h) \setminus \{x_1, \ldots, x_s\}.$$

Comparisons of Generatability

- It turns out that:

  Uniform Gen. ⇒ Non-uniform Gen. ⇒ Gen. in the limit.

- Moreover, this can be tight:

Lemma 1.

There exists classes $H_1, H_2 \subseteq \{0,1\}^X$ satisfying the UUS property such that

- H_1 is gen. in the limit but not non-uniformly gen.
- H_2 is non-uniformly gen. but not uniformly gen.

Summary of existing results

Theorem 3 and 4 (KM [2024])

Let $H \subseteq \{0,1\}^X$ satisfy the UUS property.

- If H is countable, then H is generatable in the limit.
- If H is finite, then H is uniformly generatable.

Summary of existing results

- Unfortunately, KM [2024] do not provide a characterization of which classes are uniformly and non-uniformly generatable.
- In fact, they don't provide a characterization of which classes are generatable in the limit!
- We aim to close some of these gaps by answering:

What are necessary and sufficient conditions for a class H to be uniformly or non-uniformly generatable?

Towards a characterization of generatability

- In learning theory, such conditions are often expressed in terms of combinatorial dimensions.
- A combinatorial dimension is a function

$$\dim : 2^{\{0,1\}^X} \to \mathbb{N} \cup \infty$$

  such that $\dim(H)$ captures the expressivity of H.
- For example, the VC/Littlestone dimension characterizes PAC/online learnability of a class $H \subseteq \{0,1\}^X$.
- In this work, we present a new dimension called the Closure dimension.

Closure Dimension

Definition 1(Closure Dimension)

The Closure dimension of $H \subseteq \{0,1\}^X$, denoted $C(H)$, is the largest $d \in \mathbb{N}$ for which there exists distinct $x_1, \ldots, x_d \in X^d$ such that $S(H, x_{1:d}) \geq 1$ and

$$\left| \bigcap_{h \in S(H, x_{1:d})} \operatorname{supp}(h) \right| < \infty$$

where $S(H, x_{1:d}) \coloneqq \{h \in H : x_{1:d} \subset \operatorname{supp}(h)\}$.
If this is true for arbitrarily large $d \in \mathbb{N}$, then we say $C(H) = \infty$. If it is not true for $d = 1$, then we say $C(H) = 0$.

Closure Dimension

- If $C(H) = d$, you can predict perfectly after observing any $d + 1$ distinct instances since

$$\left| \bigcap_{h \in S(H, x_{1:d+1})} \operatorname{supp}(h) \right| = \infty.$$

- If $C(H) = \infty$, the adversary can force a mistake at arbitrarily large $t \in \mathbb{N}$ since for every $t \in \mathbb{N}$, there exists $x_1, \ldots, x_n$ such that

$$\left| \bigcap_{h \in S(H, x_{1:n})} \operatorname{supp}(h) \setminus \{x_1, \ldots, x_n\} \right| = 0.$$

A Characterization of Uniform Generatability

Theorem 5.

Let $H \subseteq \{0,1\}^X$ satisfy the UUS property. The following statements are equivalent.

- H is uniformly generatable
- $C(H) < \infty$

Improvements over KM [2024]

- KM [2024] showed that all finite classes are uniformly generatable.
- We improve upon this result by giving an uncountably infinite class that is uniformly generatable.

Example 1.
Let $X = \mathbb{Z}$ and take $H = \{x \mapsto 1\{x \leq 0 \; or \; x \in A\} : A \in 2^{\mathbb{N}}\}$.
Then, H is uncountably large, satisfies the UUS property, and is uniformly generatable.

- In fact, Lemma 1 shows that countableness is not necessary for generatability in the limit!

What about Non-uniform Generatability?

We can use the Closure dimension to also provide a characterization of non-uniform generatability.

Theorem 6.

Let $H \subseteq \{0,1\}^X$ satisfy the UUS property. The following statements are equivalent.

- H is non-uniformly generatable.
- There exists a countable sequence $H_1, H_2, \ldots$ such that $H = \bigcup_{i \in \mathbb{N}} H_i$ and $C(\bigcup_{i=1}^{n} H_i) < \infty$ for all $n \geq 1$.

What about non-uniform generatability?

Theorem 6 implies that all countable classes are non-uniformly generatable!

Corollary 1.
Let $H \subseteq \{0,1\}^X$ satisfy the UUS property. If H is countable, then H is non-uniformly generatable.

This also provides an improvement* over KM [2024] since

Non-uniformly generatable ⇒ Generatable in the limit.

What about generatability in the limit?

Open Question.
What characterizes generatability in the limit?

Uniform Generation vs. Prediction

- In prediction, we are given an instance $x \in X$, and the goal is accurately predict its true label $y \in \{0, 1\}$.

- We can measure the predictability of a hypothesis class through PAC and online learnability.

- In particular, a class $H \subseteq \{0, 1\}^X$ is PAC/online learnable if and only if its the VC/Littlestone dimensions are finite.

Uniform Generation vs. Prediction

Surprisingly, we show that these two notions are incompatible.

Theorem 7.
There exists countable classes $H_1, H_2 \subseteq \{0,1\}^X$ satisfying the UUS property such that:

- $VC(H_1) = \infty$ but $C(H_1) = 0$.
- $L(H_2) = 2$ but $C(H_2) = \infty$.

Generation and Prediction are different paradigms in machine learning.

Summary

- We formalized old and new notions of generatability in the language of learning theory.
- We strengthen the results of KM [2024] by showing that all countable classes are non-uniformly generatable.
- By taking a learning-theoretic lens, we uncover fundamental differences between prediction and generation.

Extensions and Future Directions

We have barely scratched the surface.

1. Randomized Generatability: a randomized generator is a map $G: X^\star \to \Pi(X)$, where $\Pi(X)$ denotes the set of all measures. What is the right notion of randomized generatability?
2. Agnostic Generatability: what is the right model to account for the fact that we may not observe a perfect enumeration of positive instances?
3. Generatability + "X": which classes are generatable privately, fairly, robustly, …?
4. Probabilistic Generatability: what characterizes the probabilistic version of our setting where positive instances are drawn iid?

Extensions and Future Directions

5. Prompted Generatability: how do we account for prompts in our model? Will the characterization of generatability change?
6. Boosting for Generatability: given a weak, randomized generator, is it possible to "boost" it into to a strong, randomized generator?
7. Distributed Generatability: how much communication is needed to generate effectively if positive examples are distributed amongst N parties?
8. and many, many more…

Thanks for listening!

Questions?

References

1. Gold M. Language Identification in The Limit. *Information and Control*, 1967.
2. Angluin D. Finding patterns common to a set of strings. *Symposium on Theory of Computation*, 1979.
3. Angluin D. Inductive inference of formal languages from positive data. *Information and Control*, 1980.
4. Yang Y, Piantadosi S. One model for the learning of languages. *Proceedings of National Academy of Sciences*, 1988.
5. Kleinberg J, Mullainathan S. Generation in the Limit. *arXiv*, 2024.
6. Lu J, Raman V, Tewari A. Generation through the lens of learning theory. *arXiv*, 2024.