Generation through the lens of learning theory

Jiaxun Li
joint work with Vinod Raman and Ambuj Tewari
Department of Statistics, University of Michigan

Growing interest in generative AI

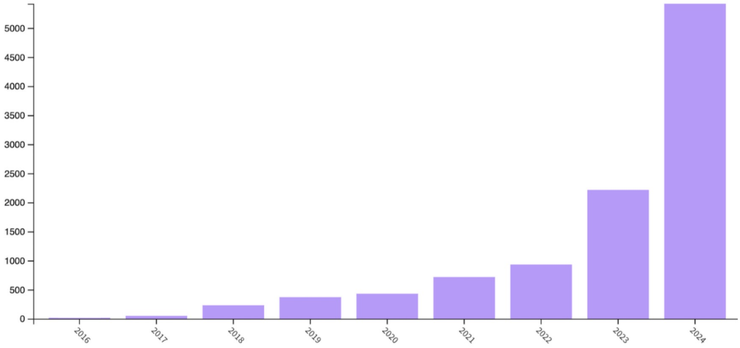

Figure: Web of Science search for “generative AI”(all fields)

Description of gen AI

- (Wiki) Generative artificial intelligence is a subset of artificial intelligence that uses generative models to produce text, images, videos, or other forms of data. These models learn the underlying patterns and structures of their training data and use them to produce new data based on the input, which often comes in the form of natural language prompts.
- (MIT news article) Generative AI can be thought of as a machine-learning model that is trained to create new data, rather than making a prediction about a specific dataset. A generative AI system is one that learns to generate more objects that look like the data it was trained on.

Basic idea of generation

- Gen AI needs training data.
- Gen AI produces new data.
- Gen AI produces data that looks like training data.

Our contribution

- We formalize generation as a sequential two-player game between an adversary and a generator. The formulation generalizes the notion of "language generation in the limit" from Kleinberg and Mullainathan [2024].
- We introduce two new paradigms of generation called "uniform" and "non-uniform" generation and provide their characterizations.
- We uncover fundamental differences between generation and prediction.

Generation model

- Formalize [Kleinberg and Mullainathan, 2024] into learning theory.
- $\mathcal{X}$ be an abstract countable instance space
- $\mathcal{H} \subset \{0,1\}^{\mathcal{X}}$ be a collection of functions that maps instances to a binary label $\{0,1\}$
- For every $h \in \mathcal{H}$, define $\text{supp}(h) = \{x \in \mathcal{X} : h(x) = 1\}$.

Generation model

- $\mathcal{X} = \mathbb{N}$.
- For each n, define h_n such that $\text{supp}(h_n) = \{n, 2n, 3n, \ldots\}$
- $\mathcal{H} = \{h_n : n = 1, 2, \ldots\}$

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**...**
h_1	1	2	3	4	5	6	7	8	...
h_2		2		4		6		8	...
h_3			3			6			
h_4				4				8	...

Figure: Example of a hypothesis Class

Generative model

- Instance space $\mathcal{X}$; Hypothesis class $\mathcal{H}$.
- Learner knows $\mathcal{H}$.

Game

Adversary chooses $h^\star \in \mathcal{H}$ and an enumeration of supp(h^*)(without learner's knowledge). At time $t = 1, 2, \ldots$

- Adversary reveals $x_t \in \text{supp}(h^*)$
- Learner generates $\hat{x}_t \in \mathcal{X}$ such that $\hat{x}_t \notin \{x_1, \ldots, x_t\}$

Goal

Learner "generates" successfully if there is a finite time t^* s.t. for all $t \geq t^*$, $\hat{x}_t \in \text{supp}(h^*)$ (assume $|\text{supp}(h^*)| = \infty$)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**...**
h_1	1	2	3	4	5	6	7	8	...
h_2		2		4		6		8	...
h_3			3			6			
h_4				4				8	...

For example, if adversary picks up h_2, and reveals $4, 8, 12, \ldots$. The learner can always generate a multiple of 4 to win.

Three variants of the model

- Gen in the limit: t^* can depend on h^* and the enumeration. [Kleinberg and Mullainathan, 2024]
- Non-uniform gen: t^* can depend on h^* but not on the enumeration.
- Uniform gen: t^* depends only on $\mathcal{H}$ [Kleinberg and Mullainathan, 2024]

Uniform Gen $\Rightarrow$ Non-uniform Gen $\Rightarrow$ Gen in the limit

Key results

- Introduce Closure Dimension $C(\mathcal{H})$,
- Both uniform and non-uniform generation can be characterized by Closure Dimension,
- Closure Dimension is also related to gen in the limit.

Version space

- Suppose learner has observed $x_1, \ldots, x_n$
- The "version space" is then

$$\mathcal{H}_{(x_{1:n})} := \{h \in \mathcal{H} : \{x_1, \ldots, x_n\} \subseteq \text{supp}(h)\}$$

- If $x \in \text{supp}(h)$ for every h in the version space, it is safe to play it

Closure dimension

That is, it is safe to play from

$$\bigcap_{h \in \mathcal{H}_{(x_{1:n})}} \operatorname{supp}(h)$$

Closure

$$\langle x_{1:n} \rangle_{\mathcal{H}} = \begin{cases} \perp & \text{if version space empty} \\ \bigcap_{h \in \mathcal{H}_{(x_{1:n})}} \operatorname{supp}(h) & \text{otherwise} \end{cases}$$

Closure dimension

- $\mathcal{X} = \mathbb{N}$, $\mathrm{supp}(h_n) = \{n, 2n, 3n, \dots\}$

Observed Instances	**Version Space** $\mathcal{H}_{(\cdot)}$	**Closure** $\langle\cdot\rangle_{\mathcal{H}}$
4	$\{h_1, h_2, h_4\}$	$\{4, 8, 12, \dots\}$
4, 6	$\{h_1, h_2\}$	$\{2, 4, 6, \dots\}$
4, 6, 5	$\{h_1\}$	$\mathbb{N}$

Closure dimension

- As $n \uparrow$, the version space shrinks and $\langle x_{1:n} \rangle_{\mathcal{H}}$ expands; once closure is infinite, learner can win
- Consider the smallest non-negative integer d such that, for every $x_1, \ldots, x_{d+1}$, either the version space is empty or the closure is infinite
  - We call d the closure dimension of $\mathcal{H}$

Key results

- $\mathcal{H}$ is uniform gen $\Leftrightarrow$ $\mathrm{C}(\mathcal{H}) < \infty$
- Finite class has finite closure dimension.

Corollary

Any finite class is uniformly generatable [Kleinberg and Mullainathan, 2024].

Key results

- $\mathcal{H}$ is non-uniform gen $\Leftrightarrow$ There exists $\mathcal{H}_i, i = 1, 2, \ldots$ with $\mathrm{C}(\mathcal{H}_i) < \infty$ s.t. $\mathcal{H}_i \uparrow \mathcal{H}$

Corollary

Any countable class is non-uniformly generatable.

- Corollary strictly improves Kleinberg and Mullainathan [2024] first main result
- Independently proved by Charikar and Pabbaraju [2024]

Key results

- $\mathcal{H}$ is gen in the limit $\Leftarrow$ There exists $\mathcal{H}_i, i = 1, \ldots, n$ with $\mathrm{C}(\mathcal{H}_i) < \infty$ s.t. $\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{H}_i$

Theorem

There exists a countable sequence $\mathcal{H}_1, \mathcal{H}_2, \ldots$ such that $\mathrm{C}(\mathcal{H}_i) = 0$ for all $i \in \mathbb{N}$ but $\bigcup \mathcal{H}_i$ is not generatable in the limit

Open Question

What characterizes generatability in the limit?

Generation vs Prediction

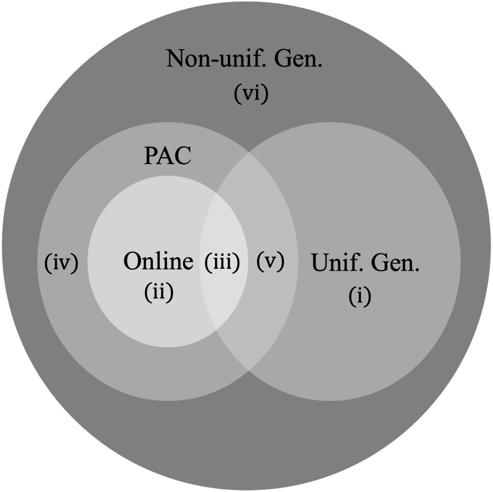

Figure: Prediction is incomparable with generation

Generation vs Prediction

- The best evidence of this difference is their behavior under unions.
- Prediction is well-behaved under unions: if $\mathcal{H}_1$ and $\mathcal{H}_2$ are PAC/online learnable then so is $\mathcal{H}_1 \cup \mathcal{H}_2$.
- This is not true for generation.

Lemma

There exists a class $\mathcal{H}$ and a hypothesis h such that $\mathrm{C}(\mathcal{H}) = 0$ but $\mathcal{H} \cup \{h\}$ is not non-uniformly generatable.

More follow-up work in the K&M model

- We also study prompted generation.
- [Charikar and Pabbaraju, 2024] study non-uniform generation, validity vs breadth and additional feedback.
- [Kalavasis et al., 2024b] and [Kalavasis et al., 2024a] study tradeoffs between validity and breadth.
- [Raman and Raman, 2025] study generation with noisy examples.
- [Peale et al., 2025] require outputs are distribution that are representative of input data.

Future Directions

- Stochastic Generatability: what characterizes generatability when positive examples are drawn iid from a distribution?
- Private Generatability: how does generatability change if one requires generators to be differentially private?
- Continuous Generatability: What if the instance space is no longer a discrete space, (e.g., the Euclidean space), how could we define generatability?

Open problems

- What characterizes generatablity in the limit?
- Is generatability in the limit closed under finite unions?
- Is the finite union of non-uniformly generatable classes generatable in the limit?
  - Solved by [Hanneke et al., 2025], the answer is no!

Theorem [Hanneke et al., 2025]

There exists classes $\mathcal{H}_1$ and $\mathcal{H}_2$ that are both non-uniformly generatable, but $\mathcal{H}_1 \cup \mathcal{H}_2$ is not generatable in the limit.

Thanks for listening!

References

M. Charikar and C. Pabbaraju. Exploring facets of language generation in the limit. *arXiv preprint arXiv:2411.15364*, 2024.

S. Hanneke, A. Karbasi, A. Mehrotra, and G. Velegkas. On union-closedness of language generation. 2025. URL `https://arxiv.org/abs/2506.18642`.

A. Kalavasis, A. Mehrotra, and G. Velegkas. Characterizations of language generation with breadth. 2024a. URL `https://arxiv.org/abs/2412.18530`.

A. Kalavasis, A. Mehrotra, and G. Velegkas. On the limits of language generation: Trade-offs between hallucination and mode collapse. *arXiv preprint arXiv:2411.09642*, 2024b.

J. Kleinberg and S. Mullainathan. Language generation in the limit. *arXiv preprint arXiv:2404.06757*, 2024.

C. Peale, V. Raman, and O. Reingold. Representative language generation. 2025. URL `https://arxiv.org/abs/2505.21819`.

A. Raman and V. Raman. Generation from noisy examples. 2025. URL `https://arxiv.org/abs/2501.04179`.