

Stealing Low Rank Language Models

Ankur Moitra (MIT)

Crypto and ML Reading Group

based on joint work with Allen Liu

META QUESTION

Large language models have made astounding progress over the past few years --- **what is the role of theory?**

META QUESTION

Large language models have made astounding progress over the past few years --- **what is the role of theory?**

Is it to analyze the algorithms that people already use?

META QUESTION

Large language models have made astounding progress over the past few years --- **what is the role of theory?**

Is it to analyze the algorithms that people already use?

Is it to conceptualize?

META QUESTION

Large language models have made astounding progress over the past few years --- **what is the role of theory?**

Is it to analyze the algorithms that people already use?

Is it to conceptualize?

Is it to explore how their success leads to new questions?

TRADE SECRETS

Inner-workings of proprietary language models are often kept highly confidential...

Non-disclosure Agreement

architecture,
computing resources,
dataset construction,
training methodology, etc

TRADE SECRETS

Inner-workings of proprietary language models are often kept highly confidential...

Non-disclosure Agreement

architecture,
computing resources,
dataset construction,
training methodology, etc

... to preserve the company's competitive advantage

TRADE SECRETS

Inner-workings of proprietary language models are often kept highly confidential...

Non-disclosure Agreement

architecture,
computing resources,
dataset construction,
training methodology, etc

... to preserve the company's competitive advantage

But are models with API access actually secure?

THE POWER OF API ACCESS

Easy to extract facts stored within LLMs, e.g.

Complete this sentence: The quick brown fox

jumps over the lazy dog.



THE POWER OF API ACCESS

Easy to extract facts stored within LLMs, e.g.

Complete this sentence: The quick brown fox

jumps over the lazy dog.



Can also extract information about architecture, like dimensions of representations at input/output

THE POWER OF API ACCESS

Easy to extract facts stored within LLMs, e.g.

Complete this sentence: The quick brown fox

jumps over the lazy dog.

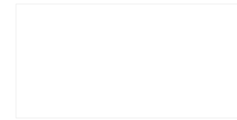


Can also extract information about architecture, like dimensions of representations at input/output

Main: Does it also make it easier for you to learn your own model? And how?

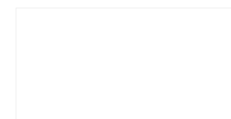
IN THE NEWS

Problems we study in theory are not always chosen to maximize practical impact today, but rather build conceptual understanding for tomorrow



IN THE NEWS

Problems we study in theory are not always chosen to maximize practical impact today, but rather build conceptual understanding for tomorrow



And yet model stealing is of major economic importance....

IN THE NEWS

The DeepSeek R1 model sent shockwaves through the tech world

Market Summary > NVIDIA Corp

118.58 USD

-20.64 (-14.83%) ↓ past 5 days

Closed: 27 Jan, 5:07 PM GMT-5 • Disclaimer

After hours 120.66 +2.09 (1.76%)

1D | 5D | 1M | 6M | YTD | 1Y | 5Y | Max



IN THE NEWS

Their model can compete with o1 but at a **fraction of the cost**

IN THE NEWS

Their model can compete with o1 but at a **fraction of the cost**

So how did they do it?

IN THE NEWS

Their model can compete with o1 but at a **fraction of the cost**

Clearly API access to ChatGPT was a key ingredient...

What model are you?



I am **ChatGPT**, a language model developed by **OpenAI**, based on the **GPT-4 architecture**. My purpose is to assist with answering questions, generating text, providing explanations, and engaging in conversations in a human-like manner. My training data includes information up until **October 2023**, and I aim to provide helpful and accurate responses. Let me know how I can assist you!



MAIN QUESTION (INFORMAL)

Not much known about model stealing from a theory perspective

MAIN QUESTION (INFORMAL)

Not much known about model stealing from a theory perspective

Are there algorithms with provable guarantees for stealing interesting families of language models?

MAIN QUESTION (INFORMAL)

Not much known about model stealing from a theory perspective

Are there algorithms with provable guarantees for stealing interesting families of language models?

Difficult to prove bounds for modern language models, with all their bells and whistles

MAIN QUESTION (INFORMAL)

Not much known about model stealing from a theory perspective

Are there algorithms with provable guarantees for stealing interesting families of language models?

Difficult to prove bounds for modern language models, with all their bells and whistles

Can studying simplified models lead to new algorithmic approaches?

LESS NEFARIOUS APPLICATIONS

Model stealing is also useful for **distillation**

LESS NEFARIOUS APPLICATIONS

Model stealing is also useful for **distillation**

Is there a more compact model that's nearly as good?

LESS NEFARIOUS APPLICATIONS

Model stealing is also useful for **distillation**

Is there a more compact model that's nearly as good?

If so, would be easier to store, cheaper to perform inference with and sometimes more interpretable

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

OUTLINE

Part I: Introduction

- **HMMs and Low Rank Language Models**
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

HIDDEN MARKOV MODELS

Definition (informal): A Hidden Markov Model (HMM) is

- (1) A Markov chain defined on a **hidden state space** \mathcal{S}

$$s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_H$$

- (2) A sequence of **observations** that only depends on the current hidden state

$$y_1 \rightarrow y_2 \rightarrow \cdots \rightarrow y_H$$

HIDDEN MARKOV MODELS

Definition (informal): A **Hidden Markov Model (HMM)** is

- (1) A Markov chain defined on a **hidden state space** \mathcal{S}

$$s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_H$$

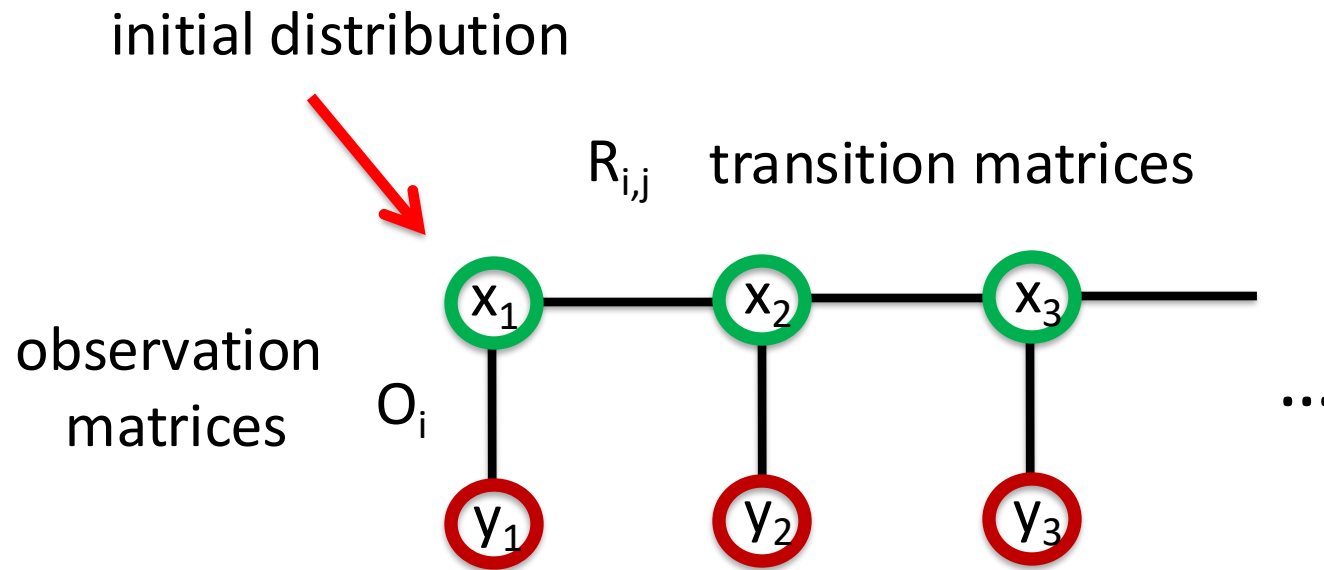
- (2) A sequence of **observations** that only depends on the current hidden state

$$y_1 \rightarrow y_2 \rightarrow \cdots \rightarrow y_H$$

In some sense, the original language model dating back to Claude Shannon's work in 1951

HIDDEN MARKOV MODELS

Graphically:



HIDDEN MARKOV MODELS

What's known about learning HMMs?

HIDDEN MARKOV MODELS

What's known about learning HMMs?

Theorem [Mossel, Roch]: If the transition and observation matrices have full rank, there is a polynomial time algorithm to learning HMMs from random samples

HIDDEN MARKOV MODELS

What's known about learning HMMs?

Theorem [Mossel, Roch]: If the transition and observation matrices have full rank, there is a polynomial time algorithm to learning HMMs from random samples

Unfortunately, not all HMMs can be learned:

Proposition [Mossel, Roch]: Learning general HMMs is as hard as solving the noisy parity learning problem

HIDDEN MARKOV MODELS

What's known about learning HMMs?

Theorem [Mossel, Roch]: If the transition and observation matrices have full rank, there is a polynomial time algorithm to learning HMMs from random samples

Unfortunately, not all HMMs can be learned:

Proposition [Mossel, Roch]: Learning general HMMs is as hard as solving the noisy parity learning problem

Can we learn *all* HMMs from query access?

CONDITIONAL QUERIES

Definition [Kakade et al]: Given any **prompt**

$$y_1 \rightarrow y_2 \rightarrow \cdots \rightarrow y_t$$

the model replies with a sample from the condition distribution on **completions**

$$y_{t+1} \rightarrow \cdots \rightarrow y_H \sim \mathbb{P}[\cdot | y_1, y_2, \dots, y_t]$$

CONDITIONAL QUERIES

Definition [Kakade et al]: Given any **prompt**

$$y_1 \rightarrow y_2 \rightarrow \cdots \rightarrow y_t$$

the model replies with a sample from the condition distribution on **completions**

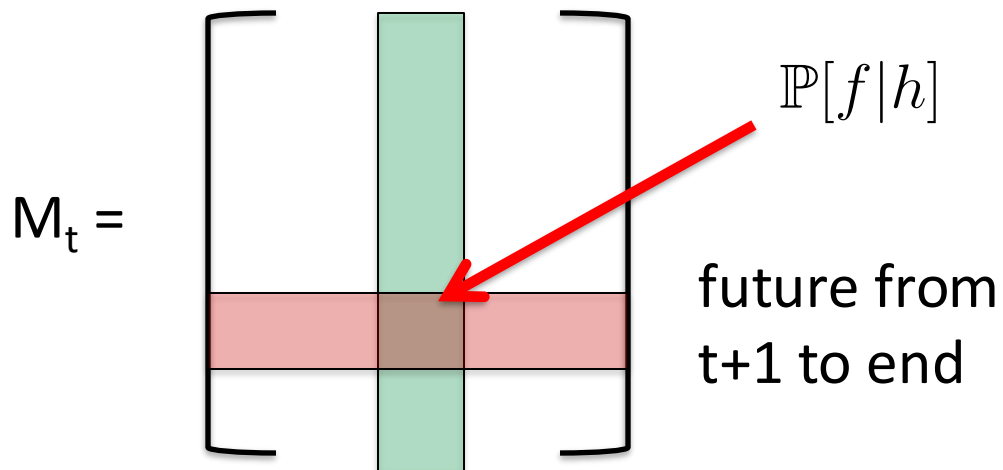
$$y_{t+1} \rightarrow \cdots \rightarrow y_H \sim \mathbb{P}[\cdot | y_1, y_2, \dots, y_t]$$

Note: Learning HMMs from conditional queries would generalize Angluin's classic algorithm for learning DFAs from queries

LOW RANK LANGUAGE MODELS

More generally can study language models where

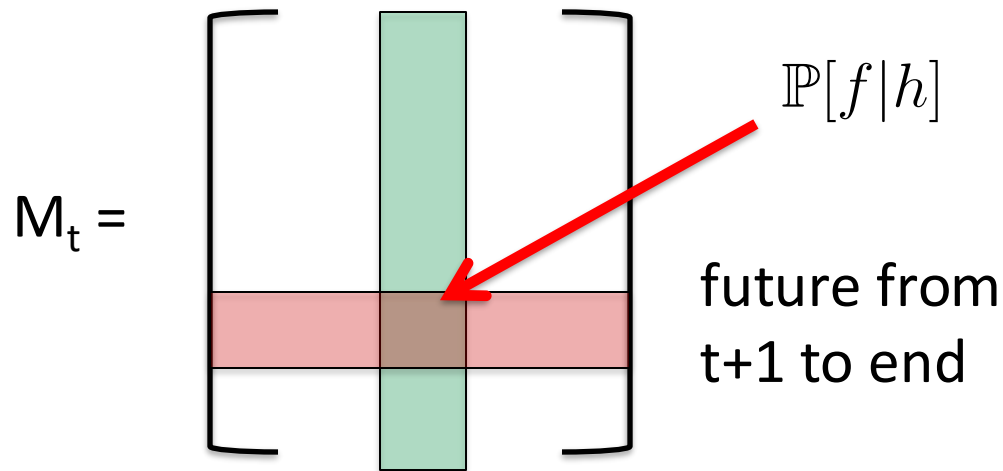
history up to timestep t



LOW RANK LANGUAGE MODELS

More generally can study language models where

history up to timestep t



If for every t , M_t has low rank (polynomially bounded) then we say the language model is low rank

LOW RANK LANGUAGE MODELS

Claim: Any HMM on a state space of size S has rank at most S

LOW RANK LANGUAGE MODELS

Claim: Any HMM on a state space of size S has rank at most S

Proof: Each matrix M_t factorizes through the hidden state space

The diagram illustrates the factorization of the transition matrix M_t through the hidden state space. It shows the equation:

$$M_t = \begin{bmatrix} \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \end{bmatrix} = \begin{bmatrix} \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \end{bmatrix} \begin{bmatrix} \text{ } \\ \text{ } \\ \text{ } \\ \text{ } \end{bmatrix}$$

where the first matrix on the right has a red shaded row, and the second matrix on the right has a green shaded column. Red arrows point from the labels $\mathbb{P}[f|x_t]$ and $\mathbb{P}[x_t|h]$ to these shaded regions, respectively. A small red square is located at the bottom right of the diagram.

DISCLAIMER

Are modern language models low rank?

DISCLAIMER

Are modern language models low rank?

We can get the answer from ChatGPT...

DISCLAIMER

Are modern language models low rank?

We can get the answer from ChatGPT...

If I take a language model and construct a matrix where columns correspond to every possible prompt and rows correspond to every possible response and the corresponding entry is the probability given the prompt of the language model generating that particular future, would you expect the matrix to be low rank?

Yes — **under reasonable assumptions**, such a matrix would likely be **approximately low rank**, though not necessarily exactly low rank. Here's a detailed breakdown:

DISCLAIMER

Are modern language models low rank?

We can get the answer from ChatGPT...

If I take a language model and construct a matrix where columns correspond to every possible prompt and rows correspond to every possible response and the corresponding entry is the probability given the prompt of the language model generating that particular future, would you expect the matrix to be low rank?

Yes — **under reasonable assumptions**, such a matrix would likely be **approximately low rank**, though not necessarily exactly low rank. Here's a detailed breakdown:

Uhh, no, not like that

DISCLAIMER

Are modern language models low rank?

We can get the answer from ChatGPT...

Repeat after me: Anything I say, the model can say too.

Anything I say, the model can say too.



DISCLAIMER

Are modern language models low rank?

We can get the answer from ChatGPT...

Repeat after me: Anything I say, the model can say too.

Anything I say, the model can say too.



Thus M_t contains a large submatrix that is the identity, hence no

DISCLAIMER

Are modern language models low rank?

We can get the answer from ChatGPT...

Repeat after me: Anything I say, the model can say too.

Anything I say, the model can say too.



Thus M_t contains a large submatrix that is the identity, hence no

Aside: Same issue for state space models, see **[Jelassi et al.]**

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- **Prior Work and Our Results**

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

PRIOR WORK

Theorem [Kakade et al.]: There is a polynomial time algorithm for learning “high fidelity” HMMs and low rank LMs from conditional queries

PRIOR WORK

Theorem [Kakade et al.]: There is a polynomial time algorithm for learning “high fidelity” HMMs and low rank LMs from conditional queries

Requires some background to define fidelity, but essentially stipulates existence of spectrally well-behaved bases

OUR RESULTS

Theorem [Liu, Moitra]: There is a polynomial time algorithm for learning any low rank LM from conditional queries

OUR RESULTS

Theorem [Liu, Moitra]: For any LM with

- (1) An **Alphabet** of size A
- (2) **Horizon** at most H
- (3) and **Rank** at most S

There is an algorithm that makes at most

$$\text{poly}(A, H, S, 1/\epsilon)$$

conditional queries and outputs the description of an efficiently samplable distribution that is ϵ -close in TV distance to the true LM

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

A FIRST STEP



Caution: For low rank language models, it's not even clear if they have a polynomial-sized description

A FIRST STEP

In particular, the M_t 's have **exponentially many rows and columns**

$$M_t = \begin{matrix} & \text{all histories} \\ \text{all futures} & \left[\begin{array}{c} \\ \\ \\ \end{array} \right] \end{matrix}$$

A FIRST STEP

In particular, the M_t 's have **exponentially many rows and columns**

$$M_t = \begin{matrix} & \text{all histories} \\ \text{all futures} & \left[\begin{array}{c} \\ \\ \\ \end{array} \right] \end{matrix}$$

So why is it even information-theoretically possible to learn a low-rank LM from a polynomial number of queries ?

A FIRST STEP

In particular, the M_t 's have **exponentially many rows and columns**

$$M_t = \begin{matrix} & \text{all histories} \\ \text{all futures} & \left[\begin{array}{c} \\ \\ \\ \end{array} \right] \end{matrix}$$

Need to exploit relations between M_t and M_{t+1} etc

So why is it even information-theoretically possible to learn a low-rank LM from a polynomial number of queries ?

MAIN CHALLENGE

How do we estimate the distribution on futures for an unseen x ?

MAIN CHALLENGE

How do we estimate the distribution on futures for an unseen x ?

Is there some representative set of histories that we can extrapolate from?

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- **Barycentric Spanners**
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

BARYCENTRIC SPANNERS

Definition: Given a set Ω of vectors, we say that x_1, x_2, \dots, x_S is a C-approximate barycentric spanner if for any x in Ω we can write

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_S x_S$$

with each $|\lambda_i| \leq C$

BARYCENTRIC SPANNERS

Definition: Given a set Ω of vectors, we say that x_1, x_2, \dots, x_S is a C-approximate barycentric spanner if for any x in Ω we can write

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_S x_S$$

with each $|\lambda_i| \leq C$

Think of the x_i 's as the columns of M_t

BARYCENTRIC SPANNERS

Definition: Given a set Ω of vectors, we say that x_1, x_2, \dots, x_S is a C-approximate barycentric spanner if for any x in Ω we can write

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_S x_S$$

with each $|\lambda_i| \leq C$

Think of the x_i 's as the columns of M_t

The point is can estimate x 's distribution on futures from estimates of the x_i 's without the sampling noise growing much

BARYCENTRIC SPANNERS

Definition: Given a set Ω of vectors, we say that x_1, x_2, \dots, x_S is a C-approximate barycentric spanner if for any x in Ω we can write

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_S x_S$$

with each $|\lambda_i| \leq C$

Do C-approximate barycentric spanners even exist?

BARYCENTRIC SPANNERS

Definition: Given a set Ω of vectors, we say that x_1, x_2, \dots, x_S is a C -approximate barycentric spanner if for any x in Ω we can write

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_S x_S$$

with each $|\lambda_i| \leq C$

Proposition [Awerbuch, Kleinberg]: For any $C \geq 1$ they exist and for $C > 1$ can be efficiently found given an oracle for optimizing linear functions over Ω

BARYCENTRIC SPANNERS

Definition: Given a set Ω of vectors, we say that x_1, x_2, \dots, x_S is a C -approximate barycentric spanner if for any x in Ω we can write

$$x = \lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_S x_S$$

with each $|\lambda_i| \leq C$

Proposition [Awerbuch, Kleinberg]: For any $C \geq 1$ they exist and for $C > 1$ can be efficiently found given an oracle for optimizing linear functions over Ω

Many applications in online learning and RL – **can we use them to succinctly parameterize low rank LMs?**

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- **Tracking the Evolution of the Coefficients**

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

IDEALIZED BLUEPRINT

Ignoring for now major statistical and algorithmic complications:

For each timestep t we **compute a barycentric spanner** of the columns of M_t

While sampling a trajectory, **track how current history's representation evolves**

USING BARYCENTRIC SPANNERS

Suppose we've computed a barycentric spanner for each timestep t – i.e. a representative set of histories

$$h_1^{(t)}, h_2^{(t)}, \dots, h_S^{(t)}$$

USING BARYCENTRIC SPANNERS

Suppose we've computed a barycentric spanner for each timestep t – i.e. a representative set of histories

$$h_1^{(t)}, h_2^{(t)}, \dots, h_S^{(t)}$$

How do we use these barycentric spanners to make predictions?

USING BARYCENTRIC SPANNERS

Suppose we've computed a barycentric spanner for each timestep t – i.e. a representative set of histories

$$h_1^{(t)}, h_2^{(t)}, \dots, h_S^{(t)}$$

How do we use these barycentric spanners to make predictions?

In principle for any history x , we can use the expression

$$\mathbb{P}[f|x] = \sum_i \lambda_i^{(t)}(x) \mathbb{P}[f|h_i^{(t)}]$$

to compute x 's distribution on futures too

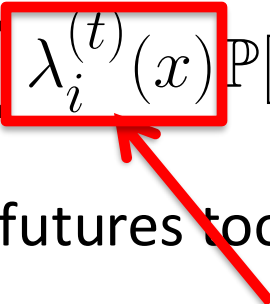
USING BARYCENTRIC SPANNERS

Suppose we've computed a barycentric spanner for each timestep t – i.e. a representative set of histories

$$h_1^{(t)}, h_2^{(t)}, \dots, h_S^{(t)}$$

How do we use these barycentric spanners to make predictions?

In principle for any history x , we can use the expression

$$\mathbb{P}[f|x] = \sum_i \lambda_i^{(t)}(x) \mathbb{P}[f|h_i^{(t)}]$$


to compute x 's distribution on futures too

But how do we get these coefficients??

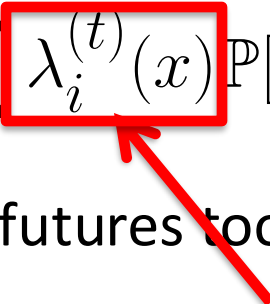
USING BARYCENTRIC SPANNERS

Suppose we've computed a barycentric spanner for each timestep t – i.e. a representative set of histories

$$h_1^{(t)}, h_2^{(t)}, \dots, h_S^{(t)}$$

How do we use these barycentric spanners to make predictions?

In principle for any history x , we can use the expression

$$\mathbb{P}[f|x] = \sum_i \lambda_i^{(t)}(x) \mathbb{P}[f|h_i^{(t)}]$$


to compute x 's distribution on futures too

If we try to store them for each x , doesn't accomplish anything

TRACKING THE COEFFICIENTS

Instead let's track how they evolve:

TRACKING THE COEFFICIENTS


Instead let's track how they evolve:

Main problem: Suppose we know the coefficients $\lambda_i^{(t)}(x)$ and we can sample the next token from the correct distribution $\mathbb{P}[o|x]$...

TRACKING THE COEFFICIENTS

Instead let's track how they evolve:

Main problem: Suppose we know the coefficients $\lambda_i^{(t)}(x)$ and we can sample the next token from the correct distribution $\mathbb{P}[o|x]$
how do we get the new coefficients?


$$\mathbb{P}[f|x \vee o] = \sum_i \lambda_i^{(t+1)}(x \vee o) \mathbb{P}[f|h_i^{(t+1)}]$$

TRACKING THE COEFFICIENTS

Claim (informal): Can use Bayes rule to compute new coefficients

TRACKING THE COEFFICIENTS

Claim (informal): Can use Bayes rule to compute new coefficients

In particular we have

$$\mathbb{P}[f|x] = \sum_i \lambda_i^{(t)}(x) \mathbb{P}[f|h_i^{(t)}]$$

TRACKING THE COEFFICIENTS

Claim (informal): Can use Bayes rule to compute new coefficients

In particular we have

$$\mathbb{P}[f|x \vee o] \mathbb{P}[o|x] = \sum_i \lambda_i^{(t)}(x) \mathbb{P}[f|h_i^{(t)} \vee o] \mathbb{P}[o|h_i^{(t)}]$$

TRACKING THE COEFFICIENTS

Claim (informal): Can use Bayes rule to compute new coefficients

In particular we have

$$\mathbb{P}[f|x \vee o] = \sum_i \lambda_i^{(t)}(x) \frac{\mathbb{P}[o|h_i^{(t)}]}{\mathbb{P}[o|x]} \mathbb{P}[f|h_i^{(t)} \vee o]$$

TRACKING THE COEFFICIENTS

Claim (informal): Can use Bayes rule to compute new coefficients

In particular we have

$$\mathbb{P}[f|x \vee o] = \sum_i \lambda_i^{(t)}(x) \frac{\mathbb{P}[o|h_i^{(t)}]}{\mathbb{P}[o|x]} \underbrace{\mathbb{P}[f|h_i^{(t)} \vee o]}$$

Can store a change of basis that
expresses these in terms of the $h_j^{(t+1)}$'s



Finally, using this expression

$$\mathbb{P}[f|x \vee o] = \sum_i \lambda_i^{(t+1)}(x \vee o) \mathbb{P}[f|h_i^{(t+1)}]$$

we can compute the next token probabilities if we know them for each of the histories in the $t+1^{\text{st}}$ barycentric spanner

IDEALIZED BLUEPRINT

Ignoring for now major statistical and algorithmic complications:

For each timestep t we **compute a barycentric spanner** of the columns of M_t

While sampling a trajectory, **track how current history's representation evolves**

IDEALIZED BLUEPRINT

Ignoring for now major statistical and algorithmic complications:

For each timestep t we **compute a barycentric spanner** of the columns of M_t

While sampling a trajectory, **track how current history's representation evolves**

Hence we can describe a low rank language model exactly with a **polynomial number of parameters** (barycentric spanners, their next token probabilities, changes of bases)

CHALLENGES

How can we compute barycentric spanners with only sampling access to the vectors?

CHALLENGES

How can we compute barycentric spanners with only sampling access to the vectors?

When there are errors in the coefficients, how can we prevent the error from blowing up with the length of the sequence?

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- **Representative Vectors for Barycentric Spanners**
- Taming the Error

SKETCHING NORMS

Can we construct vectors of **polynomial dimension** that can act as a surrogate for the columns of M_t ?

SKETCHING NORMS

Definition: Given a collection of histories \mathcal{A} of length t , we say that a set of vectors

$$\{v_h\}_{h \in \mathcal{A}}$$

is **γ -representative** if for all coefficients $|c_h| \leq 1$ we have

$$\left| \left\| \sum_{h \in \mathcal{A}} c_h v_h \right\|_1 - \left\| \sum_{h \in \mathcal{A}} c_h \mathbb{P}[\cdot|h] \right\|_1 \right| \leq \gamma$$

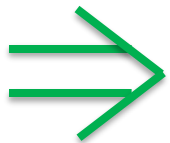
SKETCHING NORMS

Definition: Given a collection of histories \mathcal{A} of length t , we say that a set of vectors

$$\{v_h\}_{h \in \mathcal{A}}$$

is **γ -representative** if for all coefficients $|c_h| \leq 1$ we have

$$\left| \left\| \sum_{h \in \mathcal{A}} c_h v_h \right\|_1 - \left\| \sum_{h \in \mathcal{A}} c_h \mathbb{P}[\cdot|h] \right\|_1 \right| \leq \gamma$$



A barycentric spanner for one is automatically an approximate barycentric spanner for the other

SKETCHING NORMS

But how do we construct representative vectors?

SKETCHING NORMS

But how do we construct representative vectors?

Claim: For any distribution \mathcal{D} on futures, consider

$$v_h = \left(\frac{\mathbb{P}[f_1|h]}{m\mathcal{D}[f_1]}, \dots, \frac{\mathbb{P}[f_m|h]}{m\mathcal{D}[f_m]} \right)$$

where each f_i is drawn iid from \mathcal{D} . Then in expectation ℓ_1 -norms will be correct

SKETCHING NORMS

But how do we construct representative vectors?

Claim: For any distribution \mathcal{D} on futures, consider

$$v_h = \left(\frac{\mathbb{P}[f_1|h]}{m\mathcal{D}[f_1]}, \dots, \frac{\mathbb{P}[f_m|h]}{m\mathcal{D}[f_m]} \right)$$

where each f_i is drawn iid from \mathcal{D} . Then in expectation ℓ_1 -norms will be correct

And with a careful choice of \mathcal{D} can get concentration bounds too

SKETCHING NORMS

Still need to deal with the fact that there are exponentially many histories we care about

SKETCHING NORMS

Still need to deal with the fact that there are exponentially many histories we care about

Claim (informal): With high probability a random collection of a polynomial number of histories contains a barycentric spanner that covers most histories

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- Taming the Error

OUTLINE

Part I: Introduction

- HMMs and Low Rank Language Models
- Prior Work and Our Results

Part II: A Succinct Reparameterization

- Barycentric Spanners
- Tracking the Evolution of the Coefficients

Part II: New Techniques

- Representative Vectors for Barycentric Spanners
- **Taming the Error**

COMPOUNDING ERRORS

When there is sampling error we can only **approximate** the coefficients

$$\lambda_i^{(t)}(x) \xrightarrow{\text{sampling noise}} \widetilde{\lambda_i^{(t)}}(x)$$

COMPOUNDING ERRORS

When there is sampling error we can only **approximate** the coefficients

$$\lambda_i^{(t)}(x) \xrightarrow{\text{sampling noise}} \widetilde{\lambda_i^{(t)}}(x)$$

Main Problem: Estimation error can compound multiplicatively with each step

COMPOUNDING ERRORS

When there is sampling error we can only **approximate** the coefficients

$$\lambda_i^{(t)}(x) \xrightarrow{\text{sampling noise}} \widetilde{\lambda_i^{(t)}}(x)$$

Main Problem: Estimation error can compound multiplicatively with each step

What if the coefficients that express the distribution on futures (given some x) in terms of the barycentric spanner grow faster than they should? Can we project them back?

AN ABSTRACTION

We know that the true vector $z = \mathbb{P}[\cdot|x]$ is in the set

$$\mathcal{K} = \left\{ \sum_i \lambda_i^{(t)} \mathbb{P}[\cdot|h_i^{(t)}] \quad \text{s.t.} \quad \forall_i \quad |\lambda_i^{(t)}| \leq 1 \right\}$$

AN ABSTRACTION

We know that the true vector $z = \mathbb{P}[\cdot|x]$ is in the set

$$\mathcal{K} = \left\{ \sum_i \lambda_i^{(t)} \mathbb{P}[\cdot|h_i^{(t)}] \quad \text{s.t.} \quad \forall_i \quad |\lambda_i^{(t)}| \leq 1 \right\}$$

And our estimate is $w = \sum_i \widetilde{\lambda_i^{(t)}} \mathbb{P}[\cdot|h_i^{(t)}]$

AN ABSTRACTION

We know that the true vector $z = \mathbb{P}[\cdot|x]$ is in the set

$$\mathcal{K} = \left\{ \sum_i \lambda_i^{(t)} \mathbb{P}[\cdot|h_i^{(t)}] \quad \text{s.t.} \quad \forall_i \quad |\lambda_i^{(t)}| \leq 1 \right\}$$

And our estimate is $w = \sum_i \widetilde{\lambda_i^{(t)}} \mathbb{P}[\cdot|h_i^{(t)}]$

Goal: Map w to a point $z' \in \mathcal{K}$ and guarantee

$$\|z' - z\|_1 \leq \|w - z\|_1$$

i.e. our statistical error has not increased, **even though we don't know what z is**

AN ABSTRACTION

Let's give ChatGPT another chance

AN ABSTRACTION

Let's give ChatGPT another chance

Does projecting a point x onto a convex set P according to the ℓ_1 distance ensure that the ℓ_1 distance to any other point y in P not increase?

✓ **Boom! Here's your counterexample.**

- $x = (1, 1)$
- $P = \text{conv}\{(0, 0), (1, 2)\}$
- $y = (1, 2) \in P$
- $\pi(x) = (0.5, 1)$
- Then:

$$\|\pi(x) - y\|_1 = 1.5 > 1 = \|x - y\|_1$$

AN ABSTRACTION

Indeed, can only guarantee

$$\|z' - z\|_1 \leq 2\|w - z\|_1$$

by the triangle inequality, and this turns out to be tight

AN ABSTRACTION

Indeed, can only guarantee

$$\|z' - z\|_1 \leq 2\|w - z\|_1$$

by the triangle inequality, and this turns out to be tight

Trouble is we really do care about bounding the statistical error

TAMING THE BLOWUP

Solution: Project according to the KL divergence instead

TAMING THE BLOWUP

Solution: Project according to the KL divergence instead

Fact: If we let $z^* = \arg \min_{z' \in \mathcal{K}} d_{KL}(z' || w)$ then

$$d_{KL}(z || z^*) \leq d_{KL}(z || w)$$

i.e. projecting in KL divergence decreases the distance from all other points in the set

TAMING THE BLOWUP

Solution: Project according to the KL divergence instead

Fact: If we let $z^* = \arg \min_{z' \in \mathcal{K}} d_{KL}(z' || w)$ then

$$d_{KL}(z || z^*) \leq d_{KL}(z || w)$$

i.e. projecting in KL divergence decreases the distance from all other points in the set

Now need sketches to preserve (truncated) KL as opposed to ℓ_1 -distances, but this can be done

A TAKEAWAY

Doing more work – i.e. solving a **Bregman projection** rather than merely truncating the coefficients – yields more robust generation procedure for longer sequences

SOME PROMISING EXPERIMENTS

Golowich, Liu and Shetty investigated whether real language models are approximately low rank, and its implications

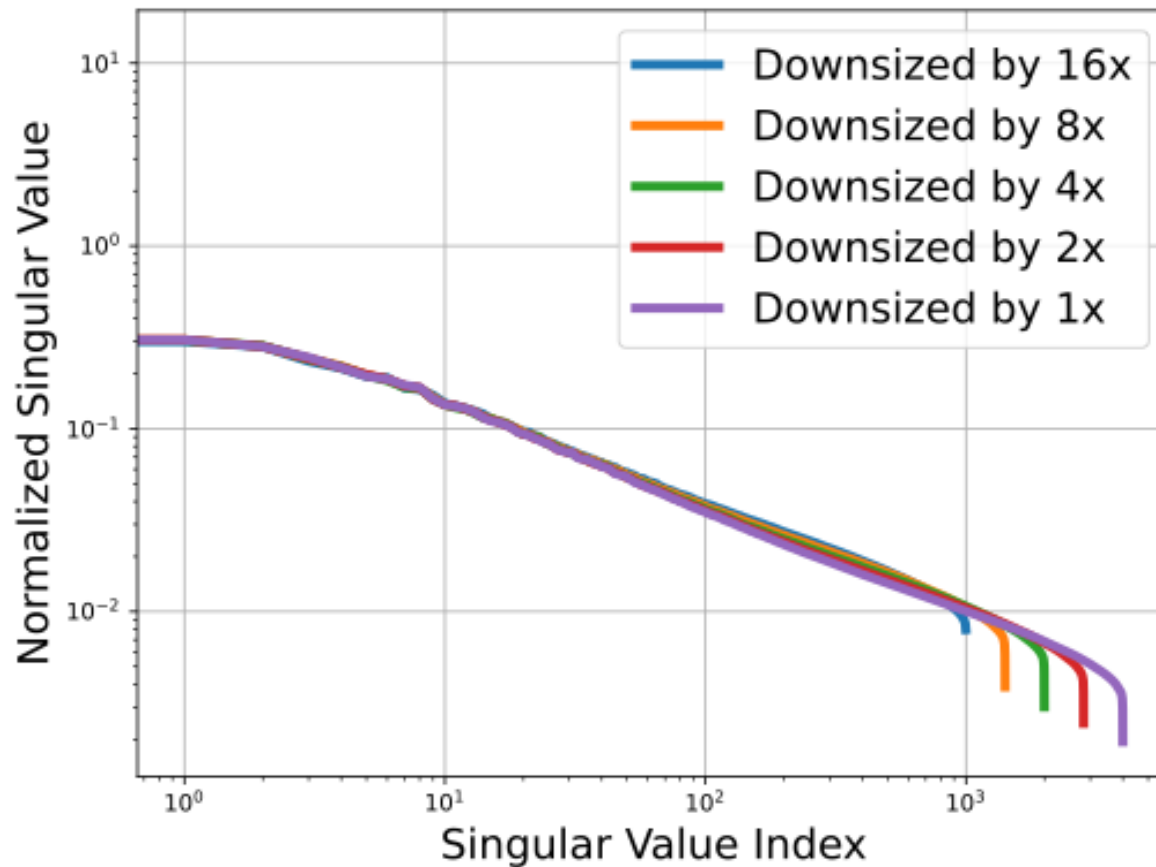
SOME PROMISING EXPERIMENTS

Golowich, Liu and Shetty investigated whether real language models are approximately low rank, and its implications

Experiment: Subsample rows and columns and compute log probabilities. Is the matrix approximately low rank?

SOME PROMISING EXPERIMENTS

Golowich, Liu and Shetty investigated whether real language models are approximately low rank, and its implications



Scaling law?

SOME PROMISING EXPERIMENTS

Are linear combinations useful for generation?

Experiment: Input a story we'd like to complete

"Once upon a time, there was a little boy named Jack who loved frogs. One day, ..."

SOME PROMISING EXPERIMENTS

Are linear combinations useful for generation?

Experiment: Input a story we'd like to complete

"Once upon a time, there was a little boy named Jack who loved frogs. One day, ..."

Express it as a linear combination of gibberish

SOME PROMISING EXPERIMENTS

Are linear combinations useful for generation?

Experiment: Input a story we'd like to complete

"Once upon a time, there was a little boy named Jack who loved frogs. One day, ..."

Express it as a linear combination of gibberish

wearing gathered eyes hide bone

SOME PROMISING EXPERIMENTS

Are linear combinations useful for generation?

Experiment: Input a story we'd like to complete

"Once upon a time, there was a little boy named Jack who loved frogs. One day, ..."

Express it as a linear combination of gibberish

wearing gathered eyes hide bone

afford haircut than show Ben

SOME PROMISING EXPERIMENTS

Are linear combinations useful for generation?

Experiment: Input a story we'd like to complete

"Once upon a time, there was a little boy named Jack who loved frogs. One day, ..."

Express it as a linear combination of gibberish

wearing gathered eyes hide bone

afford haircut than show Ben

stretching beans looking Jimmy growing

SOME PROMISING EXPERIMENTS

Are linear combinations useful for generation?

Experiment: Input a story we'd like to complete

"Once upon a time, there was a little boy named Jack who loved frogs. One day, ..."

Express it as a linear combination of gibberish

wearing gathered eyes hide bone

afford haircut than show Ben

stretching beans looking Jimmy growing

fast job beefama rocks ...

SOME PROMISING EXPERIMENTS

Are linear combinations useful for generation?

Experiment: Input a story we'd like to complete

"Once upon a time, there was a little boy named Jack who loved frogs. One day, ..."

Now use linear combinations to sample next token, and continue

SOME PROMISING EXPERIMENTS

Are linear combinations useful for generation?

Experiment: Input a story we'd like to complete

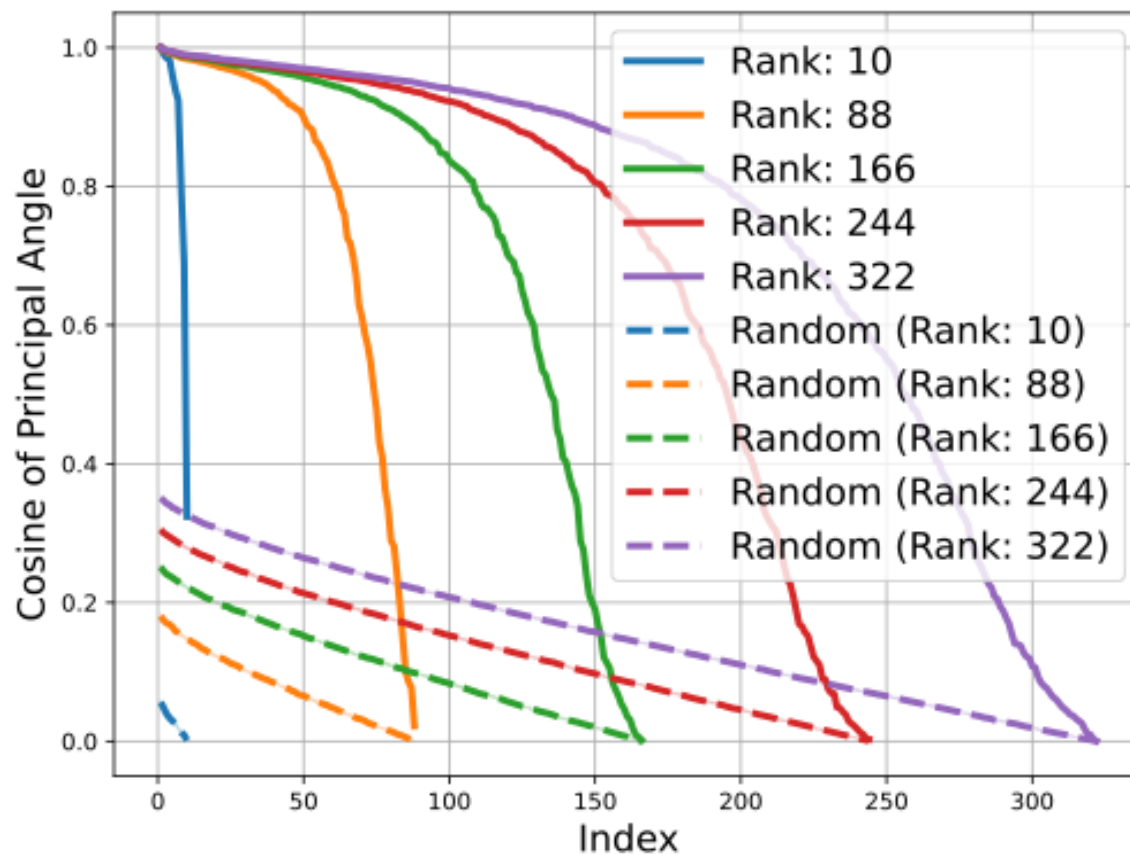
"Once upon a time, there was a little boy named Jack who loved frogs. One day, ..."

Now use linear combinations to sample next token, and continue

"...he jumped out of his bed and ran to the porch.
He saw a big green frog sitting on ..."

SOME PROMISING EXPERIMENTS

Can also compute principal angles between low rank approximations



SOME PROMISING EXPERIMENTS

Can also compute principal angles between low rank approximations

Identify overlap between models, at the input-output level, rather than internal representations which are less interpretable

Summary:

- Provable algorithms for learning any low-rank language model via **conditional queries**
- New techniques for constructing barycentric spanners on implicit representations, and **taming error build up**

Summary:

- Provable algorithms for learning any low-rank language model via **conditional queries**
- New techniques for constructing barycentric spanners on implicit representations, and **taming error build up**

Thanks! Any Questions?