

Model Stealing for Any Low-Rank Language Model

Ankur Moitra (MIT)

LLM Workshop, March 3rd

based on joint work with Allen Liu

TRADE SECRETS

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

e.g. **architecture**, **computing resources**, **dataset construction**,
training methodology, etc

TRADE SECRETS

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

e.g. **architecture**, **computing resources**, **dataset construction**, **training methodology**, etc

Are models with API access actually secure?

TRADE SECRETS

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

e.g. **architecture**, **computing resources**, **dataset construction**, **training methodology**, etc

Are models with API access actually secure?

Is it possible to learn about their parameters, training data or duplicate aspects of their functionality?

TRADE SECRETS

Easy to learn facts stored within them

Complete this sentence: The quick brown fox

jumps over the lazy dog.



TRADE SECRETS

Easy to learn facts stored within them

Complete this sentence: The quick brown fox

jumps over the lazy dog.



Does being given API access to someone else's model provably make it easier for you to learn your own?

IN THE NEWS

The DeepSeek R1 model sent shockwaves through the tech world

IN THE NEWS

The DeepSeek R1 model sent shockwaves through the tech world

Can be trained at a **fraction of the cost...**

IN THE NEWS

The DeepSeek R1 model sent shockwaves through the tech world

Can be trained at a **fraction of the cost...**

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!



...though likely by violating OpenAI's **terms of service**

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?

DISCLAIMER

Model stealing is also useful for **distillation**

DISCLAIMER

Model stealing is also useful for **distillation**

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

DISCLAIMER

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}

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

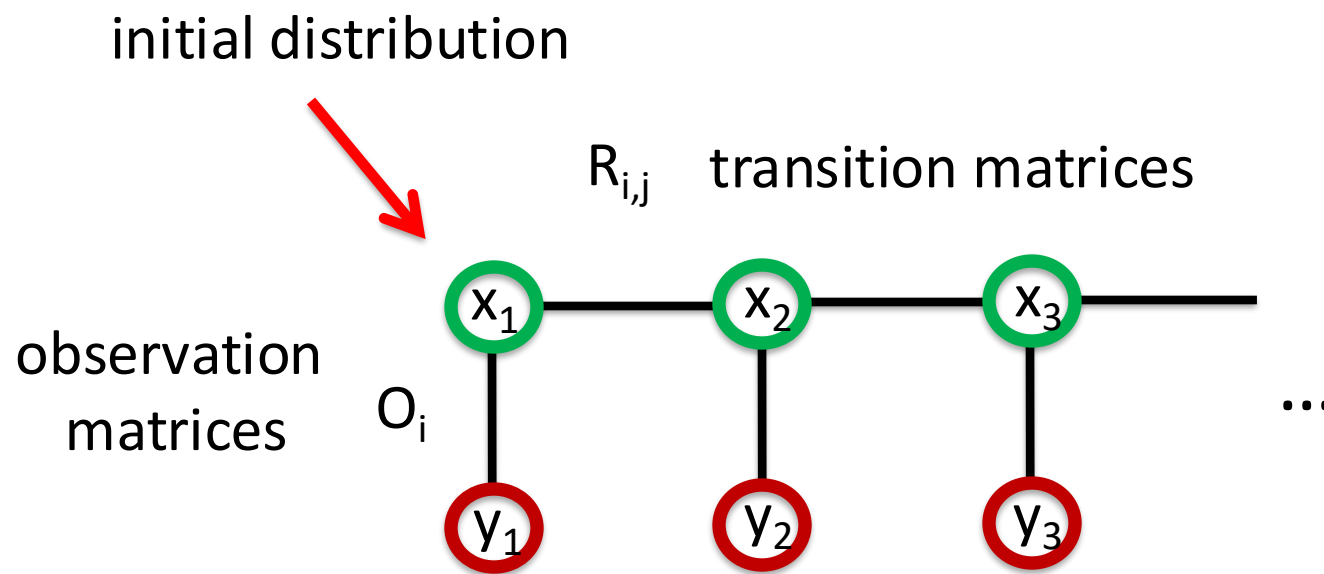
- (2) A sequence of **observations** that only depends on the current hidden state

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

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**

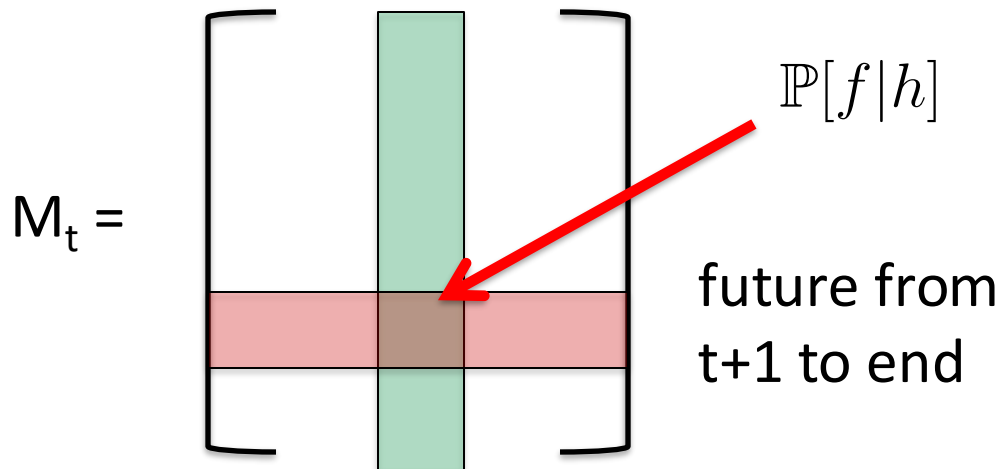
$$y_{t+1} \rightarrow \cdots \rightarrow y_H \sim \mathbb{P}[\cdot | y_1, y_2, \dots, y_t]$$

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

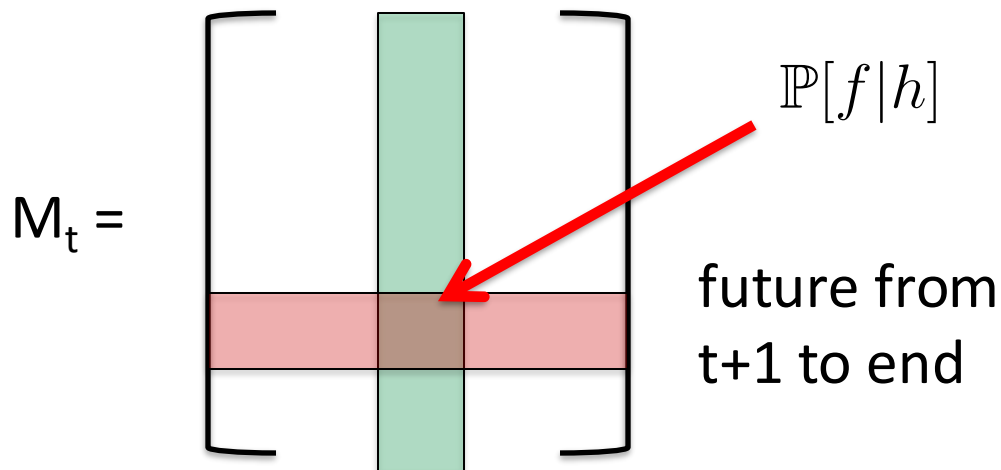
history up to timestep t



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

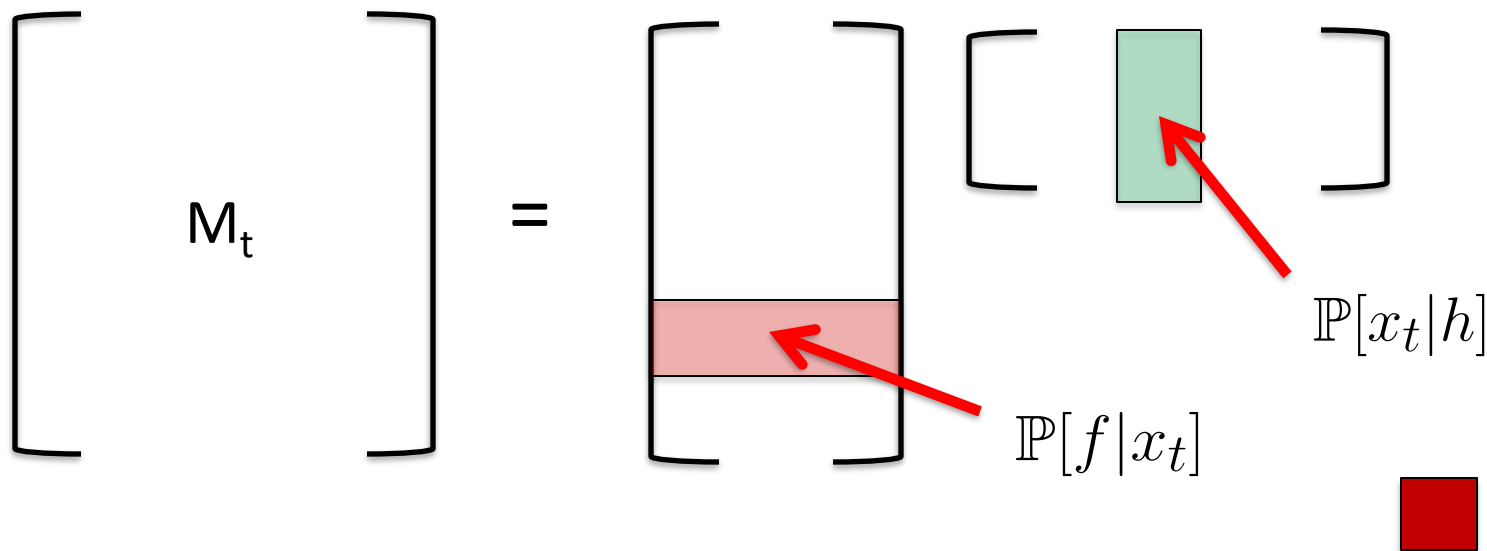
LOW RANK LANGUAGE MODELS

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



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 (INFORMAL)

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

OUR RESULTS (FORMAL)

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 model stealing is information theoretically possible

A FIRST STEP

The matrices M_t have **exponentially many rows and columns**

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

A FIRST STEP

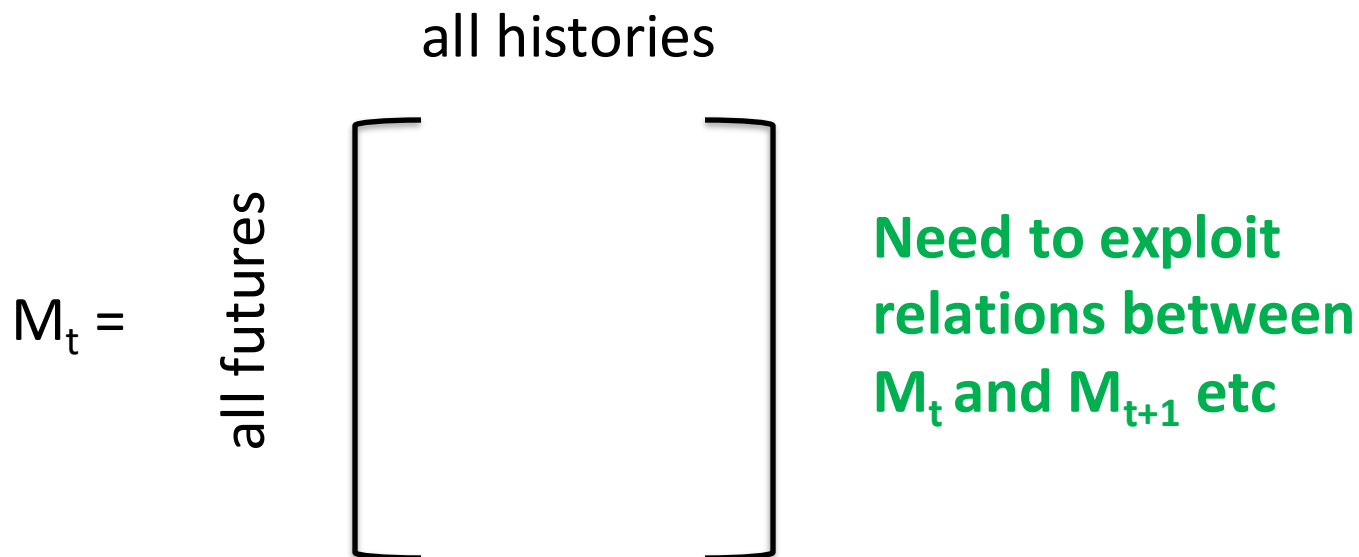
The matrices M_t have **exponentially many rows and columns**

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

Why even can we describe a low rank LM with a polynomial number of parameters?

A FIRST STEP

The matrices M_t have **exponentially many rows and columns**



Why even can we describe a low rank LM with a polynomial number of parameters?

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

Given a set Ω of vectors in an S -dimensional space how can we find a representative set?

BARYCENTRIC SPANNERS

Given a set Ω of vectors in an S -dimensional space how can we find a representative set?

Think of these vectors as columns of M_t – i.e. encoding the distribution on possible futures, given the history

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 $|\lambda_i| \leq C$

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 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 the 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??

TRACKING THE COEFFICIENTS

Main problem: Even if 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

Main problem: Even if 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

First for any future f whose $t+1^{\text{st}}$ token is o we have

$$\mathbb{P}[f|x] = \mathbb{P}[f|x \vee o]\mathbb{P}[o|x]$$

TRACKING THE COEFFICIENTS

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

First for any future f whose $t+1^{\text{st}}$ token is o we have

$$\mathbb{P}[f|x] = \mathbb{P}[f|x \vee o]\mathbb{P}[o|x]$$

Returning to our earlier expression

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

TRACKING THE COEFFICIENTS

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

First for any future f whose $t+1^{\text{st}}$ token is o we have

$$\mathbb{P}[f|x] = \mathbb{P}[f|x \vee o]\mathbb{P}[o|x]$$

Returning to our earlier expression we now have

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

TRACKING THE COEFFICIENTS

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

First for any future f whose $t+1^{\text{st}}$ token is o we have

$$\mathbb{P}[f|x] = \mathbb{P}[f|x \vee o]\mathbb{P}[o|x]$$

Returning to our earlier expression we now have

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

TRACKING THE COEFFICIENTS

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

First for any future f whose $t+1^{\text{st}}$ token is o we have

$$\mathbb{P}[f|x] = \mathbb{P}[f|x \vee o]\mathbb{P}[o|x]$$

Returning to our earlier expression we now have

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

Can compute a change of basis to express these in terms of $t+1^{\text{st}}$ barycentric spanner



And now 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 the 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 the 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 basis)

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

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

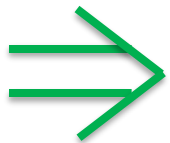
SKETCHING NORMS

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

Even though the true coefficients should be bounded (by the barycentric spanner property) the estimates might not be

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

But this is **impossible**, can only guarantee

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

by the triangle inequality, and this is tight for the ℓ_1 -projection

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

NEXT STEPS?

Sometimes can approximate language models as low rank when working with log probabilities

NEXT STEPS?

Sometimes can approximate language models as low rank when working with log probabilities

For $N = 10000$ sample histories h_i and futures f_j that are 32 tokens each and construct induced matrix M_t for **TinyStories**

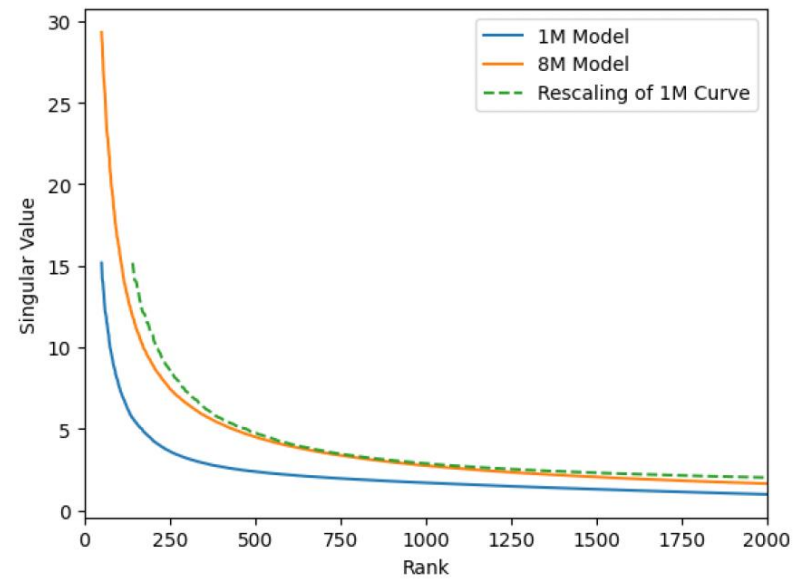
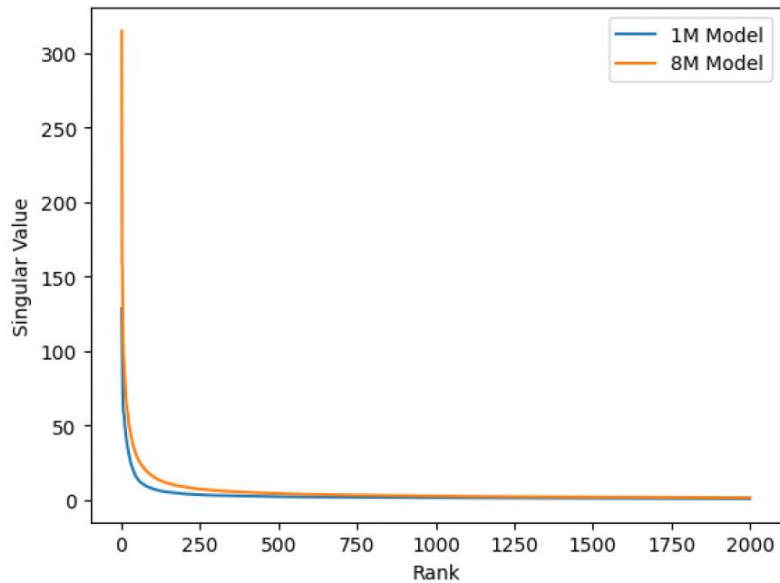
TinyStories: How Small Can Language Models Be and Still Speak Coherent English?

Ronen Eldan* and Yuanzhi Li†

Microsoft Research

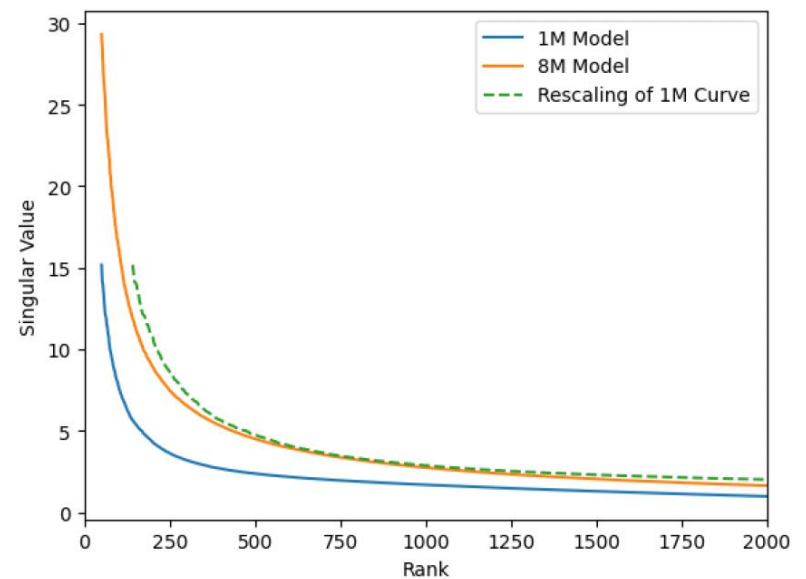
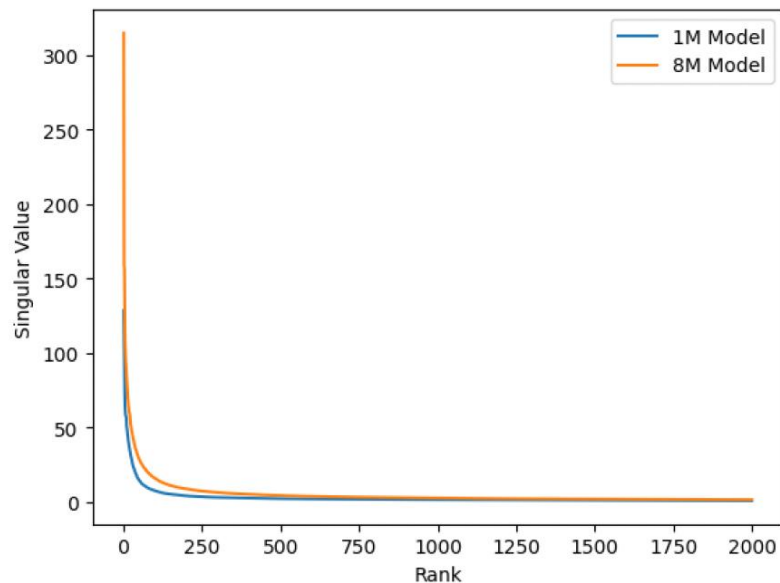
NEXT STEPS?

Plots of the singular values, appropriately scaled



NEXT STEPS?

Plots of the singular values, appropriately scaled



If you can write histories as linear combinations of other histories, what can you do with it? Reminiscent of **word embeddings**

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?