

# Low-Rank Factorizations for Fast Inference in Structured Models

Justin Chiu\* <sup>1</sup>   Yuntian Deng\* <sup>2</sup>   Alexander Rush <sup>1</sup>

<sup>1</sup>Cornell Tech

<sup>2</sup>Harvard University

October 19, 2021

# Structured Models

- ▶ Explicitly model output associations
  - ▶ Directly or through latent variables
- ▶ Focus on combinatorially large latent discrete structures
  - ▶ Complementary to continuous, deterministic representations
- ▶ More difficult to scale than alternative representations
  - ▶ Bottlenecked by time and space complexity of inference

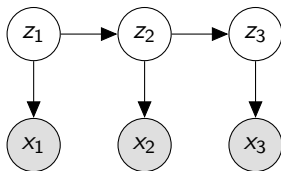
# Scaling Structured Models

- ▶ Target hypergraph models
- ▶ Impose a low-rank model constraint
  - ▶ Trades off model expressivity for cheaper inference
- ▶ Only constrain parameters used in key steps of inference

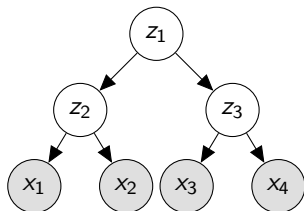
# Inference in Structured Models

- ▶ Model an observation  $x = (x_1, \dots, x_T)$  via latent structure  $z$
- ▶ Perform training and evaluation via marginalization

$$p(x) = \sum_z p(x, z)$$



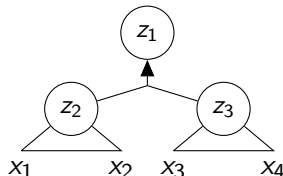
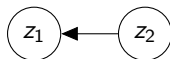
Hidden Markov models



Probabilistic context-free grammars

# Hypergraphs for Inference

- ▶ Represent dynamic programs for inference as hypergraphs
- ▶ Propagate marginals along hyperedges
  - ▶ Hyperedge consists of a head node and set of tail nodes
  - ▶ Aggregate marginals from tails to head via a matrix-vector product



Hyperedge representations for HMMs and PCFGs

# Hypergraph Marginalization

For each hyperedge  $e$  in topological order,

- ▶ Combine tail marginals  $\alpha_1, \alpha_2$  into joint tail marginal  $\beta_v$
- ▶ Apply score matrix  $\Psi_e$  and aggregate in head marginal  $\alpha_u$ 
  - ▶ Multiple hyperedges may have the same head node

---

**Algorithm 1** Hypergraph marginalization

---

**for**  $u \leftarrow v$  hyperedge  $e$  topologically **do**

$\beta_v \leftarrow \alpha_{v_1} \alpha_{v_2}^\top$   $\triangleright O(L^{|e|})$

$\alpha_u \stackrel{+}{\leftarrow} \Psi_e \beta_v$   $\triangleright O(L^{|e|+1})$

**return**  $\alpha_S^\top \mathbf{1}$

---

# Scaling with Low-rank Factorizations

- ▶ Factor matrices  $\Psi = UV^\top$ ,  $U \in \mathbb{R}^{L \times R}$ ,  $V \in \mathbb{R}^{L^{\text{el}} \times R}$

$$\boxed{\Psi} \times \boxed{\beta} = \boxed{U} \times \left( \boxed{V^\top} \times \boxed{\beta} \right)$$

- ▶ Two matrix-vector products of cost  $O(LR)$  and  $O(L^{\text{el}}R)$ 
  - ▶ Replace a factor of  $L$  with  $R$

# Low-rank Hypergraph Marginalization

---

**Algorithm 2** Low-rank marginalization

---

**for**  $u \leftarrow v_1, v_2$  hyperedge  $e$  topologically **do**

$$\beta_v \leftarrow \alpha_{v_1} \alpha_{v_2}^\top$$

$$\triangleright O(L^{|e|})$$

$$\gamma \leftarrow V_e^\top \beta_v$$

$$\triangleright O(L^{|e|}R)$$

$$\alpha_u \stackrel{+}{\leftarrow} U_e \gamma$$

$$\triangleright O(LR)$$

**return**  $\alpha_S^\top \mathbf{1}$

---



# Expressivity of Rank-constrained Models

- ▶ Rank constraints limit expressivity
- ▶ Only apply to a subset of parameters
  - ▶ Transition matrix for HMMs
  - ▶ Subset of the transition matrix for PCFGs
- ▶ Is it more expressive than a smaller model?
  - ▶ An  $L$ -state HMM with rank  $R$  ( $< L$ ) is more expressive than an  $R$ -state HMM

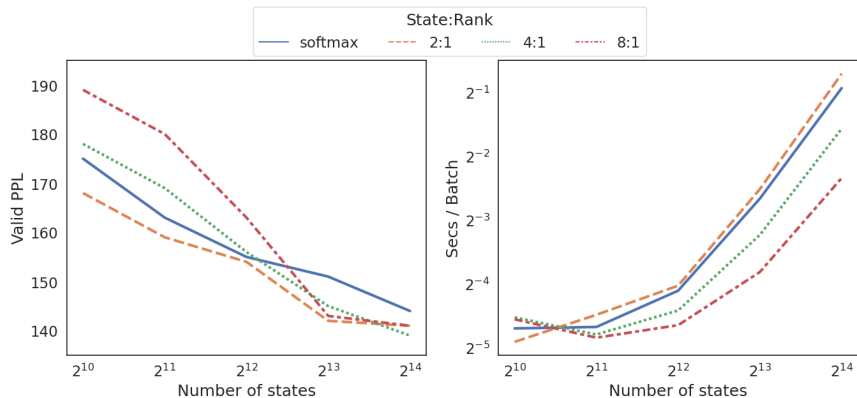
# Experiments

- ▶ Language modeling on PENN TREEBANK<sup>1</sup>
- ▶ Compare size vs speed and accuracy
  - ▶ Size = 1k to 16k state HMM, 90 to 300 state PCFG
  - ▶ Speed = Sec/Batch
  - ▶ Accuracy = Perplexity (function of likelihood)
- ▶ Unconstrained softmax HMM, PCFG vs low-rank versions

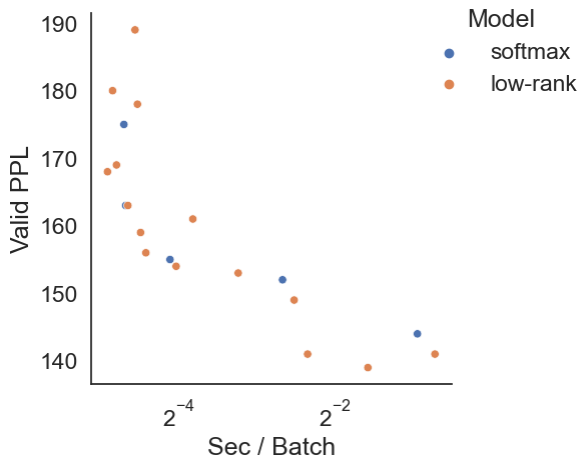
---

<sup>1</sup>Marcus, Santorini, and Marcinkiewicz, 'Building a Large Annotated Corpus of English: The Penn Treebank'.

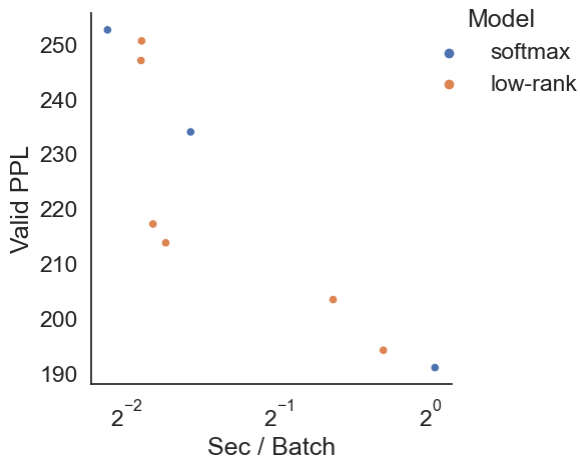
# HMM Results



# HMM Speed vs Accuracy Frontier



# PCFG Speed vs Accuracy Frontier



# Conclusion

- ▶ Introduce a low-rank factorization to speed up inference in hypergraph models
- ▶ Constrain only parameters used in inference bottlenecks
- ▶ Most effective with large models
- ▶ Please see the paper for more experiments and analysis

# Citations I



Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 'Building a Large Annotated Corpus of English: The Penn Treebank'. In: *Computational Linguistics* 19.2 (1993), pp. 313–330. URL: <https://www.aclweb.org/anthology/J93-2004>.



Zhukov, Dimitri et al. 'Cross-task weakly supervised learning from instructional videos'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3537–3545.

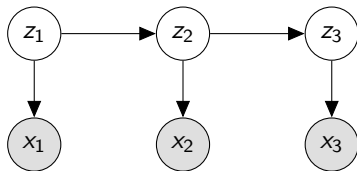
# Inference as Matrix-Vector Products

- ▶ Inference: sequence of matrix-vector products
- ▶ Speed up via fast matvec methods
- ▶ Applies to a large family of structured models



## Model 1: Hidden Markov Models (HMMs)

For times  $t$ , model states  $z_t \in [L] = \mathcal{L}$ , and tokens  $x_t \in [X] = \mathcal{X}$ ,



We wish to maximize

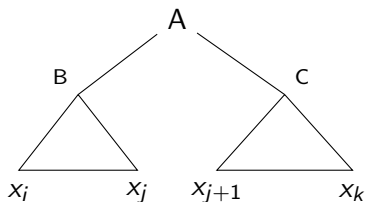
$$p(x) = \sum_{z_1} \cdots \sum_{z_T} p(x, z) = \mathbf{1}^\top \psi_1 \psi_2 \cdots \psi_T \mathbf{1},$$

where

$$\begin{aligned} [\psi_t]_{z_t, z_{t+1}} &= p(z_{t+1}, x_t \mid z_t) \\ [\psi_1]_{z_1, z_2} &= p(z_2, x_1 \mid z_1) p(z_1) \end{aligned}$$

## Model 2: Probabilistic Context-Free Grammars (PCFG)

Assign mass to each rule in a rewrite system



We wish to maximize

$$p(x) = \sum_{\text{tree: yield}(\text{tree})=x} p(\text{tree})$$

# Matvec Inference in PCFGs

For each rule define

$$[\Psi]_{z_u, (z_1, z_2)} = p(B = z_1, C = z_2 \mid A = z_u),$$

---

**Algorithm 3** PCFG Inference

---

```
for  $(i, k) \leftarrow (i, j), (j, k)$  in span-size order do  
  for  $z_1, z_2 \in \mathcal{L}_{i,j} \times \mathcal{L}_{j,k}$  do  
     $[\beta_{i,j,k}]_{(z_1, z_2)} = [\alpha_{i,j}]_{z_1} [\alpha_{j,k}]_{z_2}$   
     $\alpha_{i,k} \stackrel{+}{\leftarrow} \Psi_{i,j,k} \beta_{i,j,k}$   
return  $\alpha_{1,T}^\top \mathbf{1}$ 
```

---

# Low-Rank Factorization

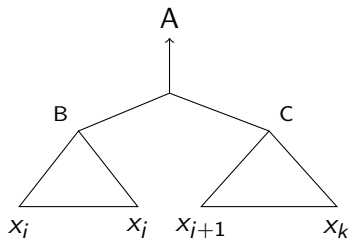
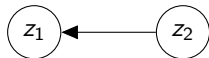
- ▶ Factor matrices  $\Psi = UV^T$ ,  $U \in \mathbb{R}^{L \times R}$ ,  $V \in \mathbb{R}^{L' \times R}$

$$\boxed{\Psi} \times \boxed{\beta} = \boxed{U} \times \left( \boxed{V^T} \times \boxed{\beta} \right)$$

- ▶ Two matrix-vector products of cost  $O(LR)$  and  $O(L'R)$

# Hypergraph Marginalization

- ▶ Models where exact inference is a directed acyclic hypergraph
- ▶ Hypergraph contains a node set and hyperedge set
  - ▶ Nodes have label set  $\mathcal{L}$
  - ▶ Hyperedges join a single head node  $u$  and a list of tail nodes  $v$



Hyperedge representations for HMMs and PCFGs

# Hypergraph Marginalization Algorithms

---

**Algorithm 4** Hypergraph marginalization

---

**for**  $u \leftarrow v$  hyperedge  $e$  topologically **do**

$$\beta_v \leftarrow \alpha_{v_1} \alpha_{v_2}^\top \quad \triangleright O(L^{|e|})$$

$$\alpha_u \stackrel{+}{\leftarrow} \Psi_e \beta_v \quad \triangleright O(L^{|e|+1})$$

**return**  $\alpha_S^\top \mathbf{1}$

---

---

**Algorithm 5** Low-rank marginalization

---

**for**  $u \leftarrow v_1, v_2$  hyperedge  $e$  topologically **do**

$$\beta_v \leftarrow \alpha_{v_1} \alpha_{v_2}^\top \quad \triangleright O(L^{|e|})$$

$$\gamma \leftarrow V_e^\top \beta_v \quad \triangleright O(L^{|e|}R)$$

$$\alpha_u \stackrel{+}{\leftarrow} U_e \gamma \quad \triangleright O(LR)$$

**return**  $\alpha_S^\top \mathbf{1}$

---

# Expressiveness and Generality

- ▶ Rank constraints limit expressivity
- ▶ Only apply to a subset of parameters
  - ▶ Transition matrix for HMMs
  - ▶ Subset of the transition matrix for PCFGs
- ▶ Is it more expressive than a smaller model?
  - ▶ An  $L$ -state HMM with rank  $R$  ( $< L$ ) is more expressive than an  $R$ -state HMM

# Experiments



# Experiments

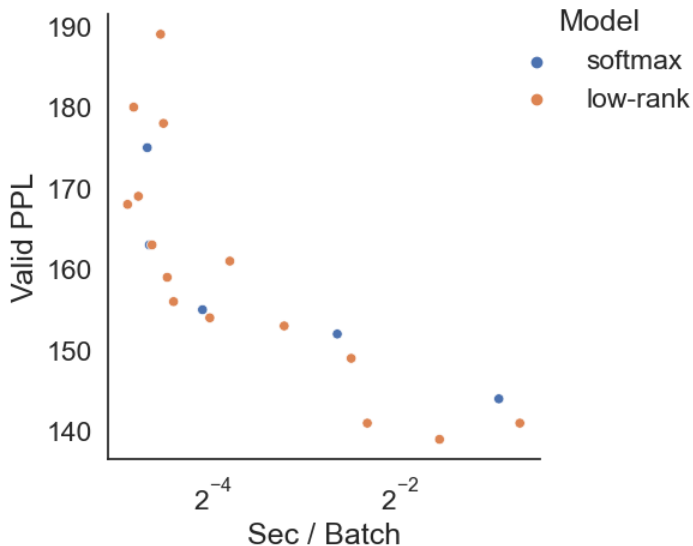
- ▶ Language modeling on PENN TREEBANK<sup>2</sup>
  - ▶ Compare speed vs accuracy frontier
  - ▶ Softmax HMM and PCFG vs low-rank versions (LHMM, LPCFG)
  - ▶ Evaluate accuracy with perplexity, a function of likelihood
- ▶ Video modeling on CROSSTASK<sup>3</sup>
  - ▶ Scale state size with fixed computation budget
  - ▶ Softmax HSMM vs low-rank HSMM
  - ▶ Evaluate accuracy with negative log-likelihood

---

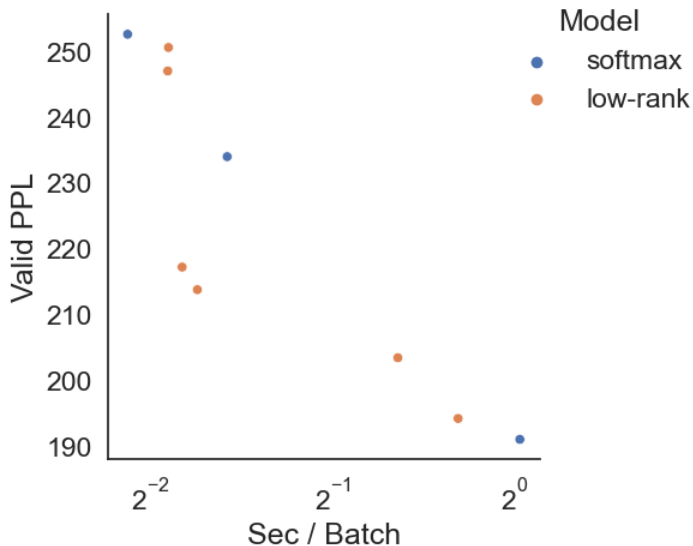
<sup>2</sup>Marcus, Santorini, and Marcinkiewicz, 'Building a Large Annotated Corpus of English: The Penn Treebank'.

<sup>3</sup>Zhukov et al., 'Cross-task weakly supervised learning from instructional videos'.

# HMM Speed vs Accuracy



# PCFG Speed vs Accuracy



# HSMM Results

Model	$L$	$N$	NLL	Sec/Batch
HSMM	$2^6$	-	1.428e5	0.78
HSMM	$2^7$	-	1.427e5	2.22
HSMM	$2^8$	-	1.426e5	7.69
LHSMM	$2^7$	$2^7$	1.427e5	4.17
LHSMM	$2^8$	$2^6$	1.426e5	5.00
LHSMM	$2^9$	$2^5$	1.424e5	5.56
LHSMM	$2^{10}$	$2^4$	1.423e5	10.00

# Conclusion

- ▶ Introduce a low-rank factorization to speed up inference
- ▶ Applies to models with hypergraph inference
- ▶ Most effective with large models

# Citations I



Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 'Building a Large Annotated Corpus of English: The Penn Treebank'. In: *Computational Linguistics* 19.2 (1993), pp. 313–330. URL: <https://www.aclweb.org/anthology/J93-2004>.



Zhukov, Dimitri et al. 'Cross-task weakly supervised learning from instructional videos'. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3537–3545.

# HMM Music Results

Model	Nott	Piano	Muse	JSB
RNN-NADE	2.31	<b>7.05</b>	<b>5.6</b>	5.19
R-Transformer	<b>2.24</b>	7.44	7.00	8.26
LSTM	3.43	7.77	7.23	8.17
LV-RNN	2.72	7.61	6.89	<b>3.99</b>
SRNN	2.94	8.20	6.28	4.74
TSBN	3.67	7.89	6.81	7.48
HMM	2.43	8.51	7.34	5.74
LHMM	2.60	8.89	7.60	5.80

## PCFG Results

$ \mathcal{N} $	$ \mathcal{P} $	Model	$N$	PPL	Sec/Batch
30	60	PCFG	-	252.60	0.29
		LPCFG	8	247.02	0.27
		LPCFG	16	250.59	0.27
60	120	PCFG	-	234.01	0.33
		LPCFG	16	217.24	0.28
		LPCFG	32	213.81	0.30
100	200	PCFG	-	191.08	1.02
		LPCFG	32	203.47	0.64
		LPCFG	64	194.25	0.81



# HSMM Speed vs Accuracy

