Low-Rank Constraints for Fast Inference in Structured Models

Justin Chiu* [1] Yuntian Deng* [2] Alexander Rush [1]

[1]Cornell Tech

[2]Harvard University

October 14, 2021

Structured Models

- Explicitly model output associations
  - Directly or through latent variables
- Focus on combinatorially large latent <u>discrete structures</u>
  - Complementary to distributed representations

Scaling Structured Models

- Prior work demonstrated: Size ↑ Performance ↑
  - Hidden Markov Models (HMM)
  - Probabilistic Context-Free Grammars (PCFG)
- Prior work scaled via
  - Sparsity for HMMs[1]
  - Low-rank tensor decompositions for PCFGs[2]
- This work: low-rank matrix constraints
  - More general
  - Less speedup

---

[1] Chiu and Rush, *Scaling Hidden Markov Language Models*.

[2] Yang, Zhao, and Tu, 'PCFGs Can Do Better: Inducing Probabilistic Context-Free Grammars with Many Symbols'.

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

Fast Matrix-Vector Products

- Matvecs take $O(L^2)$ computation
- Various fast methods
  - Sparsity (nnz entries)
  - Fast Fourier Transform ($L \log L$)
  - Low-Rank factorization (LR)
- Connected to efficient attention and kernel approximations[3]

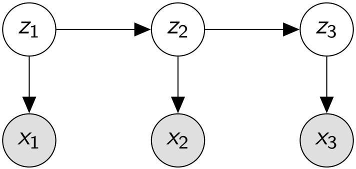
[3] Choromanski et al., *Rethinking Attention with Performers*; Peng et al., *Random Feature Attention*; Blanc and Rendle, *Adaptive Sampled Softmax with Kernel Based Sampling*.

Roadmap

- Inference in HMMs
- Inference in PCFGs
- Low-rank matvec inference
- Generalization to hypergraph inference
- Experiments

Hidden Markov Models (HMMs)

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



with joint distribution

$$p(x, z) = \prod_t p(x_t \mid z_t) p(z_t \mid z_{t-1})$$

Inference in HMMs

Given observed $x = (x_1, \ldots, x_T)$ 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

$$\begin{aligned} [\Psi_t]_{z_t, z_{t+1}} &= p(z_{t+1}, x_t \mid z_t) \\ [\Psi_1]_{z_1, z_2} &= p(z_2, x_1 \mid z_1) p(z_1) \end{aligned}$$

Probabilistic Context-Free Grammars (PCFGs)

$\mathcal{G} = (\mathcal{S}, \mathcal{N}, \mathcal{P}, \mathcal{X}, \mathcal{R})$ where

$$
\begin{aligned}
S &: \text{Start symbol} \\
\mathcal{N} &: \text{Nonterminal symbols} \\
\mathcal{P} &: \text{Preterminal symbols} \\
\mathcal{X} &: \text{Terminal symbols} \\
\mathcal{R} &: \text{Rules}
\end{aligned}
$$

Inference in PCFGs

Matvec Inference in HMMs and PCFGs

Algorithm HMM Inference

for $t \leftarrow (t+1)$ in right-to-left order **do**
  for $z_{t+1} \in \mathcal{L}$ **do**
    $[\beta_{t+1}]_{z_{t+1}} = [\alpha_{t+1}]_{z_{t+1}}$
  $\alpha_t \xleftarrow{+} \Psi_t \beta_{t+1}$
return $\alpha_0^\top \mathbf{1}$

Algorithm 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**
    $[\beta_{i,j,k}]_{(z_1,z_2)} = [\alpha_{i,j}]_{z_1}[\alpha_{j,k}]_{z_2}$
  $\alpha_{i,k} \xleftarrow{+} \Psi_{i,j,k}\beta_{i,j,k}$
return $\alpha_{1,T}^\top \mathbf{1}$

Low-Rank Factorization

- Factor matrices $\Psi \in \mathbb{R}^{L \times L}$ into product of $U, V \in \mathbb{R}^{L \times R}$

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

- Two matrix-vector products of cost $O(LR)$ each
- Also holds for rectangular Ψ

Hypergraph Marginalization

- Hypergraph represents dynamic program for exact inference
- Hyperedge consists of head node u and tail nodes $v = (v_1, v_2, \ldots)$
- PICTURE

Hypergraph Marginalization Algorithm

Algorithm Hypergraph marginalization

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

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

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

Algorithm Low-rank marginalization

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

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

$\alpha_u \xleftarrow{+} U_e \gamma$ $\qquad \triangleright\ O(LN)$

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

Experiments

Experiments

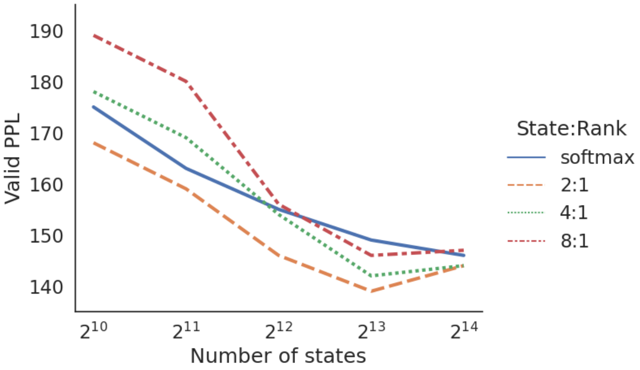
- Language modeling on PTB
- Feature map $\phi(U) = \exp(UW)$, with learned $W \in \mathbb{R}^{R \times R}$
- Baseline: Softmax HMM

HMM Accuracy

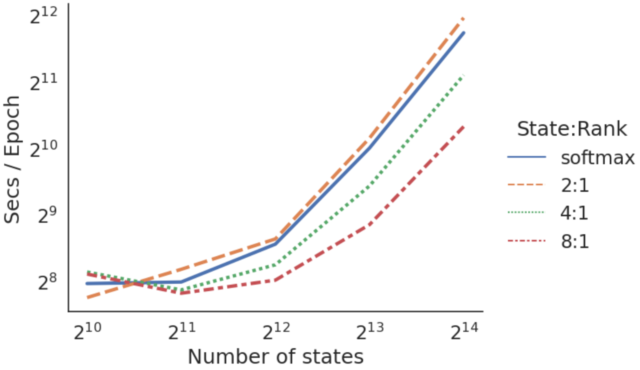
Model	Val	Test
AWD-LSTM	60.0	57.3
VL-HMM	128.6	119.5
HMM	144.3	136.8
LHMM	141.4	131.8

HMM Speed vs Accuracy Frontier

HMM Accuracy vs Rank



HMM Speed vs Rank



HMM Music Results

Model	Nott	Piano	Muse	JSB
RNN-NADE	2.31	**7.05**	**5.6**	5.19
R-Transformer	**2.24**	7.44	7.00	8.26
LSTM	3.43	7.77	7.23	8.17
LV-RNN	2.72	7.61	6.89	**3.99**
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

$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{P}\rvert$	Model	N	PPL	Batch/s
30	60	PCFG	-	252.60	4.37
		LPCFG	8	247.02	3.75
		LPCFG	16	250.59	3.74
60	120	PCFG	-	234.01	2.99
		LPCFG	16	217.24	3.55
		LPCFG	32	213.81	3.35
100	200	PCFG	-	191.08	0.98
		LPCFG	32	203.47	1.56
		LPCFG	64	194.25	1.24

HSMM Results

Model	L	N	NLL	Batch/s
HSMM	2^6	-	$1.428e5$	1.28
HSMM	2^7	-	$1.427e5$	0.45
HSMM	2^8	-	$1.426e5$	0.13
LHSMM	2^7	2^7	$1.427e5$	0.24
LHSMM	2^8	2^6	$1.426e5$	0.20
LHSMM	2^9	2^5	$1.424e5$	0.18
LHSMM	2^{10}	2^4	$1.423e5$	0.10

Citations

Blanc, Guy and Steffen Rendle. *Adaptive Sampled Softmax with Kernel Based Sampling*. 2018. arXiv: 1712.00527 [cs.LG].

Chiu, Justin T. and Alexander M. Rush. *Scaling Hidden Markov Language Models*. 2020. arXiv: 2011.04640 [cs.CL].

Choromanski, Krzysztof et al. *Rethinking Attention with Performers*. 2021. arXiv: 2009.14794 [cs.LG].

Peng, Hao et al. *Random Feature Attention*. 2021. arXiv: 2103.02143 [cs.CL].

Yang, Songlin, Yanpeng Zhao, and Kewei Tu. 'PCFGs Can Do Better: Inducing Probabilistic Context-Free Grammars with Many Symbols'. In: *CoRR* abs/2104.13727 (2021). arXiv: 2104.13727. URL: https://arxiv.org/abs/2104.13727.