Low-Rank Constraint for Fast Inference in Structured Models

Justin Chiu* and Yuntian Deng* and Alexander Rush

October 14, 2021

Structured Models

- Explicitly model output associations
  - Directly or through latent variables
- Focus on combinatorially large latent discrete structures
  - 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 mat-vecs
- Applies to a large family of structured models

Fast Matrix-Vector Products

- Mat-vecs take $O(L^2)$ computation
- Various fast methods
  - Sparsity (nnz entries)
  - Fast Fourier Transform ($L \log L$)
  - Low-Rank factorization (LR)
- Connected to work in efficient attention and low-dimensional 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

- Speeding up HMM inference
- Speeding up PCFG inference
- Generalization to hypergraph inference
- Experiments

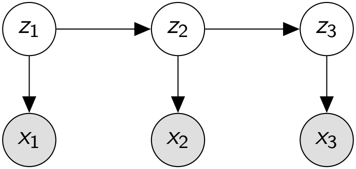
Two Examples

some text here some text here some text here some text here some text here

- Blah

Hidden Markov Models (HMMs)

For times t, model states $z_t \in [Z]$, 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

Given observed $x = (x_1, \ldots, x_T)$ We wish to maximize

$$p(x) = \sum_{z_1} \cdots \sum_{z_T} p(x, z) = \alpha_1^\top \Lambda_2 \Lambda_3 \cdots \Lambda_T \mathbf{1},$$

where we have the

start, $\quad [\alpha_1]_{z_1} = p(x_1 \mid z_1) p(z_1),$

and transition operators, $\quad [\Lambda_t]_{z_{t-1}, z_t} = p(x_t \mid z_t) p(z_t \mid z_{t-1})$

Inference: Backward Algorithm

- Performing multiplications from right to left

$$p(x) = \alpha_1^\top (\Lambda_2 (\Lambda_3 \mathbf{1}))$$

- Recursively

$$\beta_t = \Lambda_t \beta_{t+1}$$

- Requires $O(TZ^2)$ operations in total

Low-Rank Factorization

Factor matrices $\Lambda \in \mathbb{R}^{Z \times Z}$ into product of $U, V \in \mathbb{R}^{Z \times R}$

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

resulting in two matrix-vector products of cost $O(ZR)$ each

Hypergraph Marginalization

-

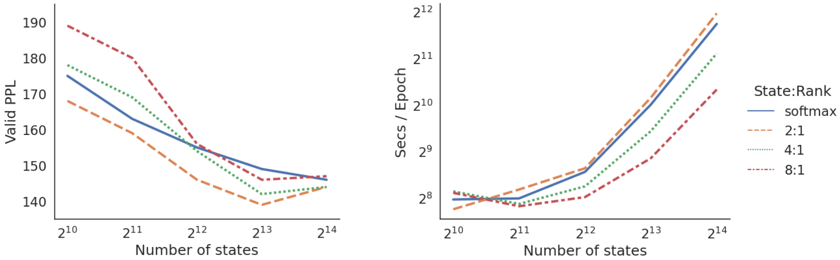
Hypergraph Marginalization Algorithm

Experiments

Experiments

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

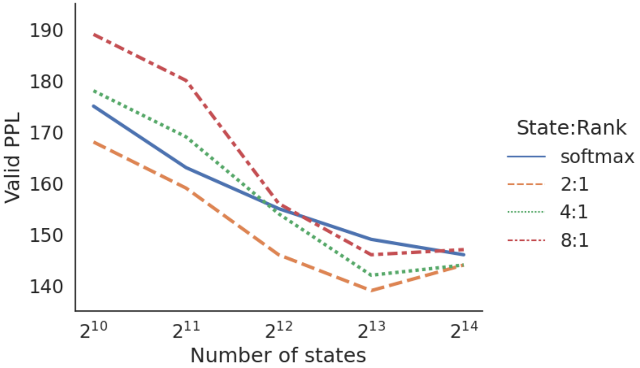
HMM Performance



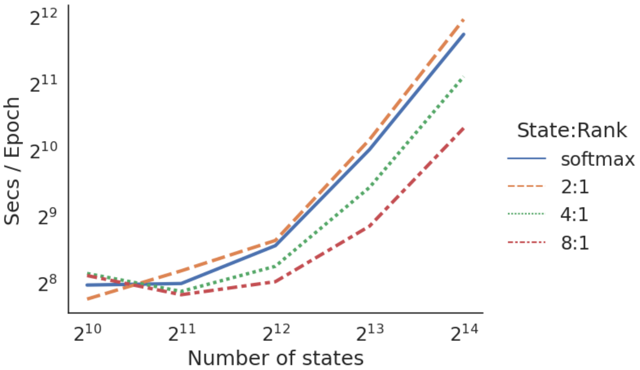
HMM Accuracy

HMM Speed vs Accuracy Frontier

HMM Accuracy vs Rank



HMM Speed vs Rank



HMM Music Results

HSMM Results

PCFG Results

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.