Low-Rank Factorizations for Fast Inference in Structured Models

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

[1]Cornell Tech

[2]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 + space complexity of marginal inference

Scaling Structured Models

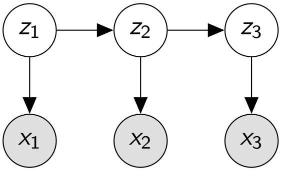
- Scaling (to the point of overparameterization) is key
- Target tractable models
  - Admit dynamic programs for exact marginalization
- Impose a low-rank model constraint
  - Trades off model expressivity for cheaper marginalization
- Only constrain parameters used in key steps of marginalization

Marginalization in Structured Models

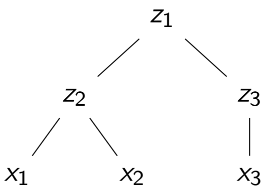
- Model an observation $x = (x_1, \ldots, x_T)$ via latent structure z
  - Latent nodes z_i
  - Nodes have discrete label set $[L]$

- Perform training and evaluation via marginalization

$$p(x) = \sum_{z} p(x, z)$$



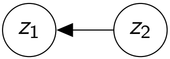
Hidden Markov models

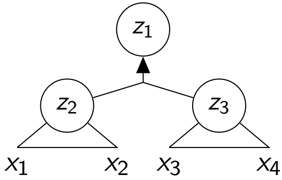


Probabilistic context-free grammars

Hypergraphs for Marginalization

- Represent marginalization dynamic programs as hypergraphs
- Hypergraphs consist of nodes and hyperedges
  - Hyperedge consists of a head node and set of tail nodes
- Perform marginalization by traversing hypergraph
  - 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 α_1, α_2 into joint tail marginal β_v
- Apply score matrix Ψ_e and aggregate in head marginal α_u
  - Matrix-vector product
  - Multiple hyperedges may have the same head node

Algorithm 1 Hypergraph marginalization / belief propagation

for $u \leftarrow v$ hyperedge e topologically **do**
  $\beta_v \leftarrow \alpha_{v_1} \alpha_{v_2}^\top$ $\triangleright$ $O(L^{|e|})$
  $\alpha_u \overset{+}{\leftarrow} \Psi_e \beta_v$ $\triangleright$ $O(L^{|e|+1})$
return $\alpha_S^\top \mathbf{1}$

Our Method: Scaling with Low-Rank Factorizations

- Hypergraph marginalization bottlenecks
  - Number of hyperedges
  - Matrix-vector product
- Approach: Impose low-rank model constraint
- Improves time and space complexity of marginalization

Low-Rank Factorizations

- Rank $R < L$ factorization
- Factor matrices $\Psi = UV^\top$, $U \in \mathbb{R}^{L\times R}$, $V \in \mathbb{R}^{L^{|e|}\times R}$

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

- Two matrix-vector products of cost $O(LR)$ and $O(L^{|e|}R)$
  - Reduced from $O(L^{|e|+1})$

Low-rank Hypergraph Marginalization

Applying the low-rank factorization,

Algorithm 2 Low-rank marginalization

for $u \leftarrow v_1, v_2$ hyperedge e topologically **do**
  $\beta_v \leftarrow \alpha_{v_1} \alpha_{v_2}^\top$ $\triangleright$ $O(L^{|e|})$
  $\gamma \leftarrow V_e^\top \beta_v$ $\triangleright$ $O(L^{|e|}R)$
  $\alpha_u \xleftarrow{+} U_e \gamma$ $\triangleright$ $O(LR)$
return $\alpha_S^\top \mathbf{1}$

- Potentially large speedups for marginalization
  - HMM from $O(L^2)$ to $O(LR)$
  - PCFG from $O(L^3)$ to $O(L^2 R)$

Expressivity of Rank-constrained Models

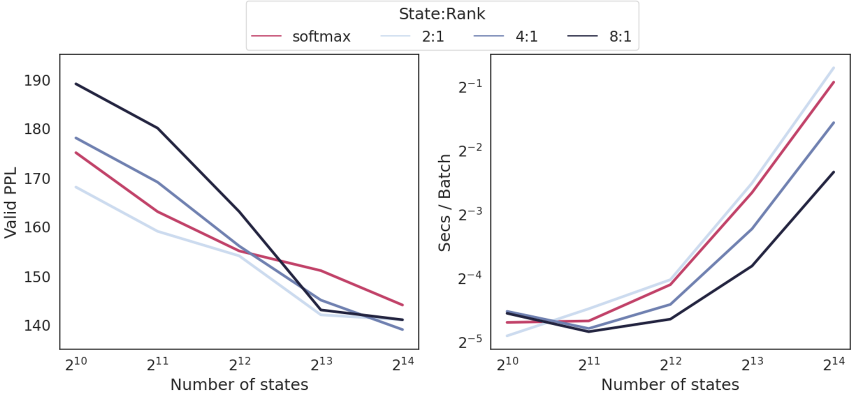
- Rank constraints limit expressivity
- Only need to constrain bottleneck parameters
  - Transition matrix for HMMs
  - Subset of the transition matrix for PCFGs
- Is it more expressive than a smaller model?
  - An L-state HMM with rank R ($< L$) is more expressive than an unconstrained R-state HMM

Experiments

- Language modeling on PENN TREEBANK[1]
- 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
- Further experiments in paper
  - Polyphonic music modeling with HMMs
  - Video modeling with HSMMs

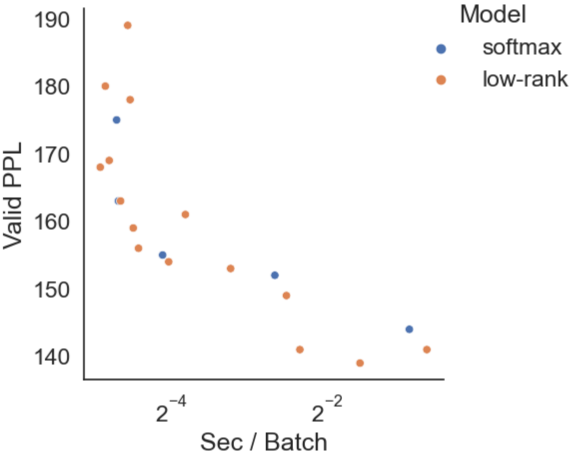
[1] Marcus, Santorini, and Marcinkiewicz, 'Building a Large Annotated Corpus of English: The Penn Treebank'.

HMM Results

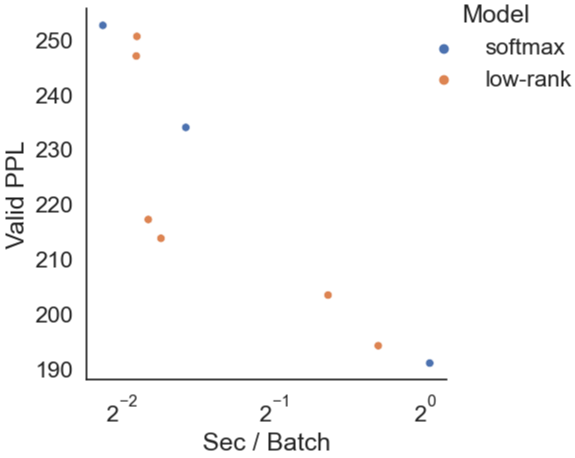


State:Rank
softmax
2:1
4:1
8:1
Valid PPL
190
180
170
160
150
140
Number of states
Secs / Batch
2^{-1}
2^{-2}
2^{-3}
2^{-4}
2^{-5}
2^{10}
2^{11}
2^{12}
2^{13}
2^{14}

HMM Speed vs Accuracy Frontier



PCFG Speed vs Accuracy Frontier



Conclusion

- Low-rank factorization speeds up marginalization
  - Constrain only bottleneck parameters
  - Most effective with large models
- Scaling improves accuracy
  - Gap with neural models still large
  - Scale further with more aggressive constraints
  - Compose with different representations
- 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.