Instructions for EMNLP 2020 Proceedings

Anonymous EMNLP submission

Abstract

todo

1 Introduction

todo

2 Background

We focus on language modeling, where we learn a distribution over sequences of tokens at the word level $\mathbf{x} = \{x_0, \dots, x_T\}$, with each token x_t an element of the finite vocabulary $\mathcal{X}$.

2.1 Hidden Markov Models

Hidden Markov Models (HMMs) specify a joint distribution over the observed words $\mathbf{x}$ and discrete latent states $\mathbf{z} = \{z_0, \dots, z_T\}$, each z_t from finite set $\mathcal{Z}$, with the following generative process: For every time step $t \in \{0, \dots, T\}$, choose a state given the previous state $z_t \mid z_{t-1}$ from the transition distribution $p(z_t \mid z_{t-1})$, where the first state z_0 is chosen from the starting distribution $p(z_0)$. Then choose a word given the current state $x_t \mid z_t$ from the emission distribution $p(x_t \mid z_t)$. This yields the joint distribution

$$p(\mathbf{x}, \mathbf{z}) = p(z_0)p(x_0 \mid z_0) \prod_{t=1}^{T} p(x_t \mid z_t)p(z_t \mid z_{t-1}) \tag{1}$$

The distributions are parameterized as follows

$$\begin{aligned} p(z_0) &\propto e^{\phi_{z_0}} \\ p(z_t \mid z_{t-1}) &\propto e^{\phi_{z_t z_{t-1}}} \\ p(x_t \mid z_t) &\propto e^{\phi_{x_t z_t}} \end{aligned} \tag{2}$$

where each $\phi_{z_0}, \phi_{z_t z_{t-1}}, \phi_{x_t z_t} \in \mathbb{R}$ may be parameterized by a scalar or a neural network.

3 Model

For language modeling, we are interested in the distribution over the observed words obtained by marginalizing over the latent states $p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$. This sum can be computed in time $O(T|\mathcal{Z}|^2)$ via dynamic programming, which becomes prohibitive if the number of latent states $|\mathcal{Z}|$ is large. However, in order to increase the capacity of HMMs, we must consider large $\mathcal{Z}$. We outline several techniques to manage the computational complexity of marginalization in HMMs.

3.1 Sparse Emission HMMs

We introduce a sparse emission constraint that allows us to efficiently perform conditional inference in large state spaces. (TODO: clarify difference in related work. should we also run this as a baseline? intuition: not enough parameter sharing if each state only emits a single word. can we prove that you need more states if the emission distribution is overconstrained?) Inspired by Cloned HMMs (Dedieu et al., 2019), we constrain each word x to be emit only by $z \in \mathcal{C}_x \subset \mathcal{Z}$:

$$p(x \mid z) \propto \begin{cases} e^{\phi_{zx}} & z \in \mathcal{C}_x \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

This allows us to perform marginalization as follows

$$\begin{aligned} p(\mathbf{x}) &= \sum_{z_0} p(z_0)p(x_0 \mid z_0) \sum_{z_1} p(z_1 \mid z_0)p(x_1 \mid z_1) \\ &\quad \cdots \sum_{z_T} p(z_T \mid z_{T-1})p(x_T \mid z_T) \\ &= \sum_{z_0 \in \mathcal{C}_{x_0}} p(z_0)p(x_0 \mid z_0) \sum_{z_1 \in \mathcal{C}_{x_1}} p(z_1 \mid z_0)p(x_1 \mid z_1) \\ &\quad \cdots \sum_{z_T \in \mathcal{C}_{x_T}} p(z_T \mid z_{T-1})p(x_T \mid z_T) \end{aligned} \tag{4}$$

which takes $O(T \max_x(|\mathcal{C}_x|)^2)$ computation.

We experiment with two constraint sets, both of which are specified as hyperparameters and remain fixed for a given model. The first constraint set is obtained by sampling subsets of states for each word i.i.d. from a uniform distribution over subsets. The second constraint set uses Brown Clusters (Brown et al., 1992) to assign all words in a given Brown cluster the same subset of states. This constraint set is advantageous, as it admits further optimizations that we discuss below.

3.2 State Dropout

We introduce a variant of dropout called state dropout that operates on the start and transition distributions. The goal of state dropout is to reduce the complexity of marginalization in a manner identical to the sparse emission constraints.

State dropout samples a mask over state partitions $\mathbf{b}_\mathcal{C} \in \{0,1\}^{|\mathcal{C}|}$ for each partition $\mathcal{C} \subset \mathcal{Z}$, such that each sampled $|\mathbf{b}_\mathcal{C}| = k$. (TODO: Mention multivariate hypergeometric?) We only apply this if the $\mathcal{C}_x$ are disjoint partitions, as with the Brown constraint set.

Let $\mathbf{b}$ be the concatenation of the $\mathbf{b}_\mathcal{C}$ for all partitions $\mathcal{C}$, such that $\mathbf{b}_z = 1$ if $\mathbf{b}_{\mathcal{C}z} = 1$, where $z \in \mathcal{C}$. The start and transition distributions with dropout are given by

$$p(z_0) \propto \begin{cases} e^{\phi_{z_0}} & \mathbf{b}_{z_0} = 1 \\ 0 & \text{otherwise} \end{cases}$$
$$p(z_t \mid z_{t-1}) \propto \begin{cases} e^{\phi_{z_t z_{t-1}}} & \mathbf{b}_{z_t} = 1 \wedge \mathbf{b}_{z_{t-1}} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Marginalizing over $\mathbf{z}$ with state dropout requires $O(Tk^2)$ computation, where k is the number of nonzero elements of $\mathbf{b}_\mathcal{C}, \forall \mathcal{C}$.

3.3 Learning

We optimize the evidence of observed sentences $\log p(\mathbf{x}) = \log \sum_\mathbf{z} p(\mathbf{x}, \mathbf{z})$ by marginalizing over latent states $\mathbf{z}$ via gradient ascent. The emission sparsity constraints we introduce as well as state dropout allow us to both perform marginalization and compute the gradient of the evidence efficiently.

4 Experiments

4.1 Language Modeling

4.1.1 Datasets

Penn Treebank

WikiText-2

4.1.2 Baselines

4.1.3 Results

Only PTB results for now.

Sparse emission constraint ablation Graph 1: Show Brown generalizes better than uniform in small model, smaller performance relative loss than uniform compared to exact HMM.

- Uniform, no dropout, 1k states (vary states per word)
- Brown, no dropout, 1k states (vary states per word)
- No sparsity, no dropout, 1k states

Graph 2: Show Brown continues to generalize better than uniform in larger states.

- Uniform, 16k states (vary states per word), unstructured dropout
- Brown, 16k states (vary states per word), unstructured dropout

Dropout ablation Graph 1: Show dropout helps but variants don't affect performance too much. Justify state dropout as the one with simplest implementation.

- Brown + unstructured dropout
- Brown + state dropout

Analysis 1: Discuss what happens with just state dropout, without partitioning.

Effective compute (k) ablation Graph 1: Fix the total number of states and examine sensitivity to the number of brown clusters / states per word.

- 16k states, 32 clusters
- 16k states, 64 clusters
- 16k states, 128 clusters

Weight tying ablation Analysis 1: Discuss performance using different tying methods (found this to not affect performance) and computational savings in terms of number of parameters. Discuss relative parameter inefficiency compared to LSTM / FF.

State size ablation Graph 1: Show performance improves as we increase the total number of states

- Brown HMM + state dropout, 16k, 128 clusters
- Brown HMM + state dropout, 32k, 128 clusters
- Brown HMM + state dropout, 64k, 128 clusters

Analysis 1: How does state usage change as clusters remain constant but the number of states (and states per word) increases?

Parameterization ablation Analysis 1: Is there a performance difference between neural / scalar parameterization, and is it consistent across state sizes?

Discussion 1: Memory savings when working with state dropout to not instantiate the full matrices during training, which makes working with a neural parameterization beneficial.

Acknowledgments

TBD

References

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479.

Antoine Dedieu, Nishad Gothoskar, Scott Swingle, Wolfgang Lehrach, Miguel Lázaro-Gredilla, and Dileep George. 2019. Learning higher-order sequential structure with cloned hmms.

A Parameter estimation

We maximize the evidence of the observed sentences $\log p(\mathbf{x}) = \log \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$ via gradient ascent.

A.1 Gradient of the evidence

Let $\psi_0(z_0, z_1) = \log p(x_0, z_0)$ and $\psi_t(z_t, z_{t+1}) = \log p(x_{t+1}, z_{t+1} \mid z_t)$ for $t \in \{1, \ldots, T-1\}$. Additionally, let $\oplus$ and $\otimes$ be addition and multiplication in the log semiring. After conditioning on the observed $\mathbf{x}$, we can express the evidence as the following:

$$A_x = \log p(\mathbf{x}) = \bigoplus_{\mathbf{z}} \bigotimes_{t=0}^{T-1} \psi_t(z_t, z_{t+1}) \tag{6}$$

where $A_{\mathbf{x}}$ is the clamped log partition function (after observing $\mathbf{x}_{0:T}$).

We show that the gradient of the log partition function with respect to the $\psi_t(z_t, z_{t+1})$ is the first moment (a general result for the cumulant generating function of exponential family distributions). Given the value of the derivative $\frac{\partial A_{\mathbf{x}}}{\partial \psi_t(z_t, z_{t+1})}$, we can then apply the chain rule to compute the total derivative with respect to downstream parameters. For example, the gradient with respect to the transition matrix of the HMM is given by

$$\frac{\partial A_{\mathbf{x}}}{\partial \theta_{ij}} = \sum_t \frac{\partial A_{\mathbf{x}}}{\partial \psi_t(i,j)} \frac{\partial \psi_t(i,j)}{\partial \theta_{ij}}$$

Recall that the derivative of logaddexp ($\oplus$ in the log semiring) is

$$\begin{aligned} \frac{\partial}{\partial x} x \oplus y &= \frac{\partial}{\partial x} \log e^x + e^y \\ &= \frac{e^x}{e^x + e^y} \\ &= \exp(x - (x \oplus y)) \end{aligned} \tag{7}$$

while the derivative of addition ($\otimes$ in the log semiring) is

$$\frac{\partial}{\partial x} x \otimes y = 1 \tag{8}$$

(TODO name variables to make this readable in 2col format) The derivative of the clamped log

partition function $A_{\mathbf{x}}$ is given by

$$
\begin{aligned}
&\frac{\partial A_{\mathbf{x}}}{\partial \psi_t(i,j)} \\
&= \frac{\partial}{\partial \psi_t(i,j)} \bigoplus_{\mathbf{z}} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \\
&= \frac{\partial}{\partial \psi_t(i,j)} \left(\left(\bigoplus_{\mathbf{z}: z_t = i, z_{t+1} = j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right. \\
&\qquad \left. \bigoplus \left(\bigoplus_{\mathbf{z}: z_t \neq i, z_{t+1} \neq j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right) \\
&= \exp \left(\left(\bigoplus_{\mathbf{z}: z_t = i, z_{t+1} = j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right. \\
&\qquad - \left(\left(\bigoplus_{\mathbf{z}: z_t = i, z_{t+1} = j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right. \\
&\qquad\qquad \left. \left. \bigoplus \left(\bigoplus_{\mathbf{z}: z_t \neq i, z_{t+1} \neq j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) \right) \right) \\
&= \exp \left(\left(\bigoplus_{\mathbf{z}: z_t = i, z_{t+1} = j} \bigotimes_{t'} \psi_{t'}(z_{t'}, z_{t'+1}) \right) - A_{\mathbf{x}} \right)
\end{aligned}
$$

which is the value at i and j of the edge marginal for z_t, z_{t+1} obtained via the forward-backward algorithm. The second equality is a result of the associativity of $\oplus$; the third equality is a result of Eqn. 7; and the fourth equality from Eqn. 8.

B Supplemental Material