Methods of Moments for Learning Stochastic Languages: Unified Presentation and Empirical Comparison
(Supplementary Material)

Borja Balle William L Hamilton Joelle Pineau

Reasoning and Learning Laboratory
School of Computer Science
McGill University
Montreal, QC, Canada

{bballe|whamil3|jpineau}@cs.mcgill.ca

1 Consistency of Tensor Decomposition Method

Let $A = \langle \boldsymbol{\alpha}_0, \boldsymbol{\alpha}_\infty, \mathbf{T}, \{\mathbf{O}_\sigma\}_{\sigma\in\Sigma}\rangle$ be a HMM with $n \leq |\Sigma|$ states. To prove the consistency of the algorithm described in Section 3.3, we will show that the when given access to data computed from $f = f_A$, the algorithm returns a HMM identical to A modulo a permutation on the states.

Suppose that we are given sets of prefixes and suffixes $\mathcal{P}, \mathcal{S} \subset \Sigma^\star$ like in the algorithm. We start by defining some notation. Let us write $\mathbf{O}_\mathcal{P} \in \mathbb{R}^{\mathcal{P}\times n}$ with rows given by $\mathbf{e}_u^\top \mathbf{O}_\mathcal{P} = \boldsymbol{\alpha}_0^\top \mathbf{A}_u$ and $\mathbf{S} \in \mathbb{R}^{\mathcal{S}\times n}$ with rows given by $\mathbf{e}_v^\top \mathbf{S} = (\mathbf{A}_v \boldsymbol{\alpha}_\infty)^\top$. For convenience we also define $\mathbf{O}_\mathcal{S} = \mathbf{S}\mathbf{T}^\top$. To prove our results we will need to make some assumption on f, $\mathcal{P}$ and $\mathcal{S}$. These are in line with usual assumptions made in the analysis of tensor-based methods of moments. Furthermore, we have reasons to believe that without such assumptions the task is stastically and computationally hard [Anandkumar et al., 2012]. Thus, we suppose that the following is satisfied.

Assumption 1. *The matrices $\mathbf{O}$, $\mathbf{O}_\mathcal{P}$ and $\mathbf{O}_\mathcal{S}$ have rank n, and for every $i \in [n]$ we have $\boldsymbol{\alpha}_\infty(i) < 1$.*

Our proof starts by obtaining expressions for the Hankel matrices and tensors computed by the algorithm in terms of the parameters of A.

Lemma 1.1. *The following are true for the Hankel matrices and tensors computed by the algorithm:*

$$\mathbf{H}_{\mathcal{P},\Sigma,\mathcal{S}} = \sum_{i\in[n]} (\mathbf{O}_\mathcal{P}\mathbf{e}_i) \otimes (\mathbf{O}\mathbf{e}_i) \otimes (\mathbf{O}_\mathcal{S}\mathbf{e}_i) \ , \tag{1}$$

$$\bar{\mathbf{H}}_{\mathcal{P},\Sigma} = \mathbf{O}_\mathcal{P}\mathbf{D}_\mathcal{S}\mathbf{O}^\top \ , \tag{2}$$

$$\bar{\mathbf{H}}_{\mathcal{P},\mathcal{S}} = \mathbf{O}_\mathcal{P}\mathbf{D}_\Sigma\mathbf{O}_\mathcal{S}^\top \ , \tag{3}$$

$$\bar{\mathbf{H}}_{\Sigma,\mathcal{S}} = \mathbf{O}_\mathcal{S}\mathbf{D}_\mathcal{P}\mathbf{O}^\top \ , \tag{4}$$

$$\mathbf{H}_{\mathcal{P},\mathcal{S}} = \mathbf{O}_\mathcal{P}\mathbf{S}^\top \ , \tag{5}$$

$$\mathbf{H}^{\mathrm{p}}_{\mathcal{P},\Sigma} = \mathbf{O}_\mathcal{P}\mathbf{O}^\top \ , \tag{6}$$

where $\mathbf{D}_\mathcal{S}, \mathbf{D}_\Sigma, \mathbf{D}_\mathcal{P} \in \mathbb{R}^{n\times n}$ are rank n diagonal matrices. Additionally, we have $\mathbf{D}_\Sigma = \mathbf{I} - \mathrm{diag}(\boldsymbol{\alpha}_\infty)$ and $\mathrm{rank}(\bar{\mathbf{H}}_{\mathcal{P},\mathcal{S}}) = n$.

Proof. The expression for $\mathbf{H}_{\mathcal{P},\Sigma,\mathcal{S}}$ follows from writing the following in tensor notation using the definitions of $\mathbf{O}_{\mathcal{P}}$, $\mathbf{O}$, and $\mathbf{O}_{\mathcal{S}}$:

$$\mathbf{H}_{\mathcal{P},\Sigma,\mathcal{S}}(u,\sigma,v) = \boldsymbol{\alpha}_0^\top \mathbf{A}_u \mathbf{A}_\sigma \mathbf{A}_v \boldsymbol{\alpha}_\infty = \boldsymbol{\alpha}_0^\top \mathbf{A}_u \mathbf{O}_\sigma \mathbf{T} \mathbf{A}_v \boldsymbol{\alpha}_\infty \ . \tag{7}$$

Now, using the expression for $\mathbf{H}_{\mathcal{P},\Sigma,\mathcal{S}}$ we get obtain the expressions for the three integrations as follows:

$$\begin{aligned}
\bar{\mathbf{H}}_{\mathcal{P},\Sigma} &= \mathbf{H}_{\mathcal{P},\Sigma,\mathcal{S}}(\mathbf{I},\mathbf{I},\mathbf{1}) && (8)\\
&= \sum_{i\in[n]} (\mathbf{O}_{\mathcal{P}}\mathbf{e}_i) \otimes (\mathbf{O}\mathbf{e}_i) \otimes (\mathbf{1}^\top \mathbf{O}_{\mathcal{S}}\mathbf{e}_i) && (9)\\
&= \sum_{i\in[n]} \mathbf{w}_{\mathcal{S}}(i)(\mathbf{O}_{\mathcal{P}}\mathbf{e}_i) \otimes (\mathbf{O}\mathbf{e}_i) && (10)\\
&= \mathbf{O}_{\mathcal{P}} \operatorname{diag}(\mathbf{w}_{\mathcal{S}}) \mathbf{O}^\top = \mathbf{O}_{\mathcal{P}} \mathbf{D}_{\mathcal{S}} \mathbf{O}^\top \ , && (11)\\
\bar{\mathbf{H}}_{\mathcal{P},\mathcal{S}} &= \mathbf{H}_{\mathcal{P},\Sigma,\mathcal{S}}(\mathbf{I},\mathbf{1},\mathbf{I}) && (12)\\
&= \sum_{i\in[n]} (\mathbf{O}_{\mathcal{P}}\mathbf{e}_i) \otimes (\mathbf{1}^\top \mathbf{O}\mathbf{e}_i) \otimes (\mathbf{O}_{\mathcal{S}}\mathbf{e}_i) && (13)\\
&= \sum_{i\in[n]} \mathbf{w}_{\Sigma}(i)(\mathbf{O}_{\mathcal{P}}\mathbf{e}_i) \otimes (\mathbf{O}_{\mathcal{S}}\mathbf{e}_i) && (14)\\
&= \mathbf{O}_{\mathcal{P}} \operatorname{diag}(\mathbf{w}_{\Sigma}) \mathbf{O}_{\mathcal{S}}^\top = \mathbf{O}_{\mathcal{P}} \mathbf{D}_{\Sigma} \mathbf{O}_{\mathcal{S}}^\top \ , && (15)\\
\bar{\mathbf{H}}_{\Sigma,\mathcal{S}} &= \mathbf{H}_{\mathcal{P},\Sigma,\mathcal{S}}(\mathbf{1},\mathbf{I},\mathbf{I}) && (16)\\
&= \sum_{i\in[n]} (\mathbf{1}^\top \mathbf{O}_{\mathcal{P}}\mathbf{e}_i) \otimes (\mathbf{O}\mathbf{e}_i) \otimes (\top \mathbf{O}_{\mathcal{S}}\mathbf{e}_i) && (17)\\
&= \sum_{i\in[n]} \mathbf{w}_{\mathcal{P}}(i)(\mathbf{O}\mathbf{e}_i) \otimes (\mathbf{O}_{\mathcal{S}}\mathbf{e}_i) && (18)\\
&= \mathbf{O} \operatorname{diag}(\mathbf{w}_{\mathcal{P}}) \mathbf{O}_{\mathcal{S}}^\top = \mathbf{O}\mathbf{D}_{\mathcal{P}} \mathbf{O}_{\mathcal{S}}^\top \ , && (19)
\end{aligned}$$

where $\mathbf{w}_{\mathcal{S}}^\top = \mathbf{1}^\top \mathbf{O}_{\mathcal{S}}$, $\mathbf{w}_{\Sigma}^\top = \mathbf{1}^\top \mathbf{O}$, and $\mathbf{w}_{\mathcal{P}}^\top = \mathbf{1}^\top \mathbf{O}_{\mathcal{P}}$. In particular, note that the HMM constraints imply that $\mathbf{w}_{\Sigma}(i) = 1 - \boldsymbol{\alpha}_\infty(i)$ for every i. Thus, we have $\mathbf{D}_{\Sigma} = \mathbf{I} - \operatorname{diag}(\boldsymbol{\alpha}_\infty)$. Since Assumption 1 guarantees that all entries in $\mathbf{w}_{\mathcal{P}}, \mathbf{w}_{\Sigma}, \mathbf{w}_{\mathcal{S}}$ are strictly positive, we see that matrices $\mathbf{D}_{\mathcal{P}}$, $\mathbf{D}_{\Sigma}$, and $\mathbf{D}_{\mathcal{S}}$ must have full rank.

For the other two Hankel matrices we see that the first expression follows immediately from $\mathbf{H}_{\mathcal{P},\mathcal{S}}(u,v) = f(uv)$, and the second one follows from

$$\begin{aligned}
\mathbf{H}^{\mathrm{p}}_{\mathcal{P},\Sigma}(u,\sigma) = f(u\sigma\Sigma^\star) &= \sum_{x\in\Sigma^\star} f(u\sigma x) && (20)\\
&= \boldsymbol{\alpha}_0^\top \mathbf{A}_u \mathbf{A}_\sigma \left(\sum_x \mathbf{A}_x \right) \boldsymbol{\alpha}_\infty && (21)\\
&= \boldsymbol{\alpha}_0^\top \mathbf{A}_u \mathbf{A}_\sigma \left(\sum_{k\geq 0} \left(\sum_\sigma \mathbf{A}_\sigma \right)^k \right) \boldsymbol{\alpha}_\infty && (22)\\
&= \boldsymbol{\alpha}_0^\top \mathbf{A}_u \mathbf{A}_\sigma \left(\sum_{k\geq 0} \left(\mathbf{I} - \operatorname{diag}(\boldsymbol{\alpha}_\infty) \right)^k \mathbf{T}^k \right) \boldsymbol{\alpha}_\infty && (23)\\
&= \boldsymbol{\alpha}_0^\top \mathbf{A}_u \mathbf{A}_\sigma \left(\mathbf{I} - \left(\mathbf{I} - \operatorname{diag}(\boldsymbol{\alpha}_\infty) \right) \mathbf{T} \right)^{-1} \boldsymbol{\alpha}_\infty && (24)\\
&= \boldsymbol{\alpha}_0^\top \mathbf{A}_u \mathbf{O}_\sigma \mathbf{T} \mathbf{1} && (25)\\
&= \boldsymbol{\alpha}_0^\top \mathbf{A}_u \mathbf{O}_\sigma \mathbf{1} \ . && (26)
\end{aligned}$$

where we used that $(\mathbf{I} - (\mathbf{I} - \operatorname{diag}(\boldsymbol{\alpha}_\infty))\mathbf{T})\mathbf{1} = \boldsymbol{\alpha}_\infty$. □

Note that because $\bar{\mathbf{H}}_{\mathcal{P},\mathcal{S}}$ has rank n, we can find $\mathbf{Q}_{\mathcal{P}} \in \mathbb{R}^{n\times\mathcal{P}}$ and $\mathbf{Q}_{\mathcal{S}} \in \mathbb{R}^{n\times\mathcal{S}}$ such that $\tilde{\mathbf{H}}_{\mathcal{P},\mathcal{S}} = \mathbf{Q}_{\mathcal{P}}\bar{\mathbf{H}}_{\mathcal{P},\mathcal{S}}\mathbf{Q}_{\mathcal{S}}$ is invertible. Using these, we define the matrices $\mathbf{N} = \mathbf{Q}_{\mathcal{S}}^{\top}\tilde{\mathbf{H}}_{\mathcal{P},\mathcal{S}}^{-1}\mathbf{Q}_{\mathcal{P}}$, $\mathbf{X}_{\Sigma} = \bar{\mathbf{H}}_{\Sigma,\mathcal{S}}\mathbf{N}\bar{\mathbf{H}}_{\mathcal{P},\Sigma}$, and the tensor $\mathbf{Y}_{\Sigma} = \mathbf{H}_{\mathcal{P},\Sigma,\mathcal{S}}(\mathbf{N}^{\top}\bar{\mathbf{H}}_{\Sigma,\mathcal{S}}^{\top}, \mathbf{I}, \mathbf{N}\bar{\mathbf{H}}_{\mathcal{P},\Sigma})$. The following result analyzes the eigenpairs found by the tensor decomposition step on the whitened version of $\mathbf{Y}_{\Sigma}$.

Lemma 1.2. *There exists $\mathbf{W} \in \mathbb{R}^{\Sigma\times n}$ such that $\mathbf{W}^{\top}\mathbf{X}_{\Sigma}\mathbf{W} = \mathbf{I}$. Furthermore, the tensor $\mathbf{Z}_{\Sigma} = \mathbf{Y}_{\Sigma}(\mathbf{W},\mathbf{W},\mathbf{W}) \in \mathbb{R}^{n\times n\times n}$ admits a robust eigendecomposition $\mathbf{Z}_{\Sigma} = \sum_{j\in[n]}\gamma_j\mathbf{z}_j^{\otimes 3}$, where for each $j \in [n]$ we have*

$$\gamma_j = \frac{1}{\sqrt{\mathbf{w}_{\mathcal{P}}(i)\mathbf{w}_{\Sigma}(i)\mathbf{w}_{\mathcal{S}}(i)}} \ , \tag{27}$$

$$\mathbf{z}_j = \sqrt{\frac{\mathbf{w}_{\mathcal{P}}(i)\mathbf{w}_{\mathcal{S}}(i)}{\mathbf{w}_{\Sigma}(i)}}\mathbf{W}^{\top}\mathbf{O}\mathbf{e}_i \ , \tag{28}$$

for some $i \in [n]$.

Proof. First we need to analyze the construction of matrix $\mathbf{X}_{\Sigma}$ and tensor $\mathbf{Y}_{\Sigma}$. By using the expressions from Lemma 1.1 we can see that $\tilde{\mathbf{H}}_{\mathcal{P},\mathcal{S}}^{-1} = (\mathbf{O}_{\mathcal{S}}^{\top}\mathbf{Q}_{\mathcal{S}}^{\top})^{-1}\mathbf{D}_{\Sigma}^{-1}(\mathbf{Q}_{\mathcal{P}}\mathbf{O}_{\mathcal{P}})^{-1}$. Therefore, we have the following decomposition for $\mathbf{X}_{\Sigma}$:

$$\begin{aligned}
\mathbf{X}_{\Sigma} &= \left(\mathbf{O}\mathbf{D}_{\mathcal{P}}\mathbf{O}_{\mathcal{S}}^{\top}\right)\left(\mathbf{Q}_{\mathcal{S}}^{\top}(\mathbf{O}_{\mathcal{S}}^{\top}\mathbf{Q}_{\mathcal{S}}^{\top})^{-1}\mathbf{D}_{\Sigma}^{-1}(\mathbf{Q}_{\mathcal{P}}\mathbf{O}_{\mathcal{P}})^{-1}\mathbf{Q}_{\mathcal{P}}\right)\left(\mathbf{O}_{\mathcal{P}}\mathbf{D}_{\mathcal{S}}\mathbf{O}^{\top}\right) && (29)\\
&= \mathbf{O}\mathbf{D}_{\mathcal{P}}\mathbf{D}_{\Sigma}^{-1}\mathbf{D}_{\mathcal{S}}\mathbf{O}^{\top} && (30)\\
&= \sum_{i\in[n]}\frac{\mathbf{w}_{\mathcal{P}}(i)\mathbf{w}_{\mathcal{S}}(i)}{\mathbf{w}_{\Sigma}(i)}(\mathbf{O}\mathbf{e}_i)\otimes(\mathbf{O}\mathbf{e}_i) \ . && (31)
\end{aligned}$$

Now we observe that because of Assumption 1, matrix $\mathbf{X}_{\Sigma}$ is positive definite of rank n. Thus there exists $\mathbf{W} \in \mathbb{R}^{\Sigma\times n}$ such that $\mathbf{W}^{\top}\mathbf{X}_{\Sigma}\mathbf{W} = \mathbf{I}$. By the decomposition of $\mathbf{X}_{\Sigma}$ given above, this implies that writing $\mathbf{w}_X(i) = \mathbf{w}_{\mathcal{P}}(i)\mathbf{w}_{\mathcal{S}}(i)/\mathbf{w}_{\Sigma}(i)$, we have

$$\mathbf{I} = \mathbf{X}_{\Sigma}(\mathbf{W},\mathbf{W}) = \sum_{i\in[n]}(\sqrt{\mathbf{w}_X(i)}\mathbf{W}^{\top}\mathbf{O}\mathbf{e}_i)\otimes(\sqrt{\mathbf{w}_X(i)}\mathbf{W}^{\top}\mathbf{O}\mathbf{e}_i) \ . \tag{32}$$

Thus, the vectors $\mathbf{x}_i = \sqrt{\mathbf{w}_X(i)}\mathbf{W}^{\top}\mathbf{O}\mathbf{e}_i \in \mathbb{R}^n$ for $i \in [n]$ form an orthonormal basis.

To obtain an expression for $\mathbf{Y}_{\Sigma}$, we first check the following two equalities:

$$\begin{aligned}
\bar{\mathbf{H}}_{\Sigma,\mathcal{S}}\mathbf{N}\mathbf{O}_{\mathcal{P}} &= \left(\mathbf{O}\mathbf{D}_{\mathcal{P}}\mathbf{O}_{\mathcal{S}}^{\top}\right)\left(\mathbf{Q}_{\mathcal{S}}^{\top}(\mathbf{O}_{\mathcal{S}}^{\top}\mathbf{Q}_{\mathcal{S}}^{\top})^{-1}\mathbf{D}_{\Sigma}^{-1}(\mathbf{Q}_{\mathcal{P}}\mathbf{O}_{\mathcal{P}})^{-1}\mathbf{Q}_{\mathcal{P}}\right)\mathbf{O}_{\mathcal{P}} && (33)\\
&= \mathbf{O}\mathbf{D}_{\mathcal{P}}\mathbf{D}_{\Sigma}^{-1} \ , && (34)\\
\mathbf{O}_{\mathcal{S}}^{\top}\mathbf{N}\bar{\mathbf{H}}_{\mathcal{P},\Sigma} &= \mathbf{O}_{\mathcal{S}}^{\top}\left(\mathbf{Q}_{\mathcal{S}}^{\top}(\mathbf{O}_{\mathcal{S}}^{\top}\mathbf{Q}_{\mathcal{S}}^{\top})^{-1}\mathbf{D}_{\Sigma}^{-1}(\mathbf{Q}_{\mathcal{P}}\mathbf{O}_{\mathcal{P}})^{-1}\mathbf{Q}_{\mathcal{P}}\right)\mathbf{O}_{\mathcal{P}}\mathbf{D}_{\mathcal{S}}\mathbf{O}^{\top} && (35)\\
&= \mathbf{D}_{\Sigma}^{-1}\mathbf{D}_{\mathcal{S}}\mathbf{O}^{\top} \ . && (36)
\end{aligned}$$

Combining these expressions with the definition of $\mathbf{Y}_{\Sigma}$ we obtain

$$\begin{aligned}
\mathbf{Y}_{\Sigma} &= \mathbf{H}_{\mathcal{P},\Sigma,\mathcal{S}}(\mathbf{N}^{\top}\bar{\mathbf{H}}_{\Sigma,\mathcal{S}}^{\top}, \mathbf{I}, \mathbf{N}\bar{\mathbf{H}}_{\mathcal{P},\Sigma}) && (37)\\
&= \sum_{i\in[n]}(\bar{\mathbf{H}}_{\Sigma,\mathcal{S}}\mathbf{N}\mathbf{O}_{\mathcal{P}}\mathbf{e}_i)\otimes(\mathbf{O}\mathbf{e}_i)\otimes(\bar{\mathbf{H}}_{\mathcal{P},\Sigma}^{\top}\mathbf{N}^{\top}\mathbf{O}_{\mathcal{S}}\mathbf{e}_i) && (38)\\
&= \sum_{i\in[n]}(\mathbf{O}\mathbf{D}_{\mathcal{P}}\mathbf{D}_{\Sigma}^{-1}\mathbf{e}_i)\otimes(\mathbf{O}\mathbf{e}_i)\otimes(\mathbf{O}\mathbf{D}_{\mathcal{S}}\mathbf{D}_{\Sigma}^{-1}\mathbf{e}_i) && (39)\\
&= \sum_{i\in[n]}\frac{\mathbf{w}_{\mathcal{P}}(i)\mathbf{w}_{\mathcal{S}}(i)}{\mathbf{w}_{\Sigma}(i)^2}(\mathbf{O}\mathbf{e}_i)\otimes(\mathbf{O}\mathbf{e}_i)\otimes(\mathbf{O}\mathbf{e}_i) \ . && (40)
\end{aligned}$$

Now we can show that the whitened version of $\mathbf{Y}_\Sigma$ has a robust orthonormal decomposition:

$$\mathbf{Z}_\Sigma = \mathbf{Y}_\Sigma(\mathbf{W}, \mathbf{W}, \mathbf{W}) \tag{41}$$

$$= \sum_{i\in[n]} \frac{\mathbf{w}_\mathcal{P}(i)\mathbf{w}_\mathcal{S}(i)}{\mathbf{w}_\Sigma(i)^2}(\mathbf{W}^\top\mathbf{O}\mathbf{e}_i) \otimes (\mathbf{W}^\top\mathbf{O}\mathbf{e}_i) \otimes (\mathbf{W}^\top\mathbf{O}\mathbf{e}_i) \tag{42}$$

$$= \sum_{i\in[n]} \frac{1}{\sqrt{\mathbf{w}_\mathcal{P}(i)\mathbf{w}_\Sigma(i)\mathbf{w}_\mathcal{S}(i)}}\mathbf{x}_i \otimes \mathbf{x}_i \otimes \mathbf{x}_i \ . \tag{43}$$

Therefore, any robust eigenpair $(\gamma, \mathbf{z})$ of $\mathbf{Z}_\Sigma$ must be of the form $\gamma = (\mathbf{w}_\mathcal{P}(i)\mathbf{w}_\Sigma(i)\mathbf{w}_\mathcal{S}(i))^{-1/2}$ and $\mathbf{z} = \mathbf{x}_i$ for some $i \in [n]$. □

Next we show that the algorithm recovers weighted and permuted versions of the parameters of the target HMM. Recall that each column of the matrix $\tilde{\mathbf{O}} \in \mathbb{R}^{\Sigma\times n}$ corresponds to $\gamma(\mathbf{W}^\top)^+\mathbf{z}$ for some robust eigenpair $(\gamma, \mathbf{z})$ of $\mathbf{Z}_\Sigma$. Thus, since

$$\gamma(\mathbf{W}^\top)^+\mathbf{z} = \frac{1}{\sqrt{\mathbf{w}_\mathcal{P}(i)\mathbf{w}_\Sigma(i)\mathbf{w}_\mathcal{S}(i)}}(\mathbf{W}^\top)^+(\sqrt{\mathbf{w}_X(i)}\mathbf{W}^\top\mathbf{O}\mathbf{e}_i) \tag{44}$$

$$= \frac{1}{\mathbf{w}_\Sigma(i)}\mathbf{O}\mathbf{e}_i \tag{45}$$

for some $i \in [n]$, we get $\tilde{\mathbf{O}} = \mathbf{O}\mathbf{D}_\Sigma^{-1}\mathbf{\Pi}$ for some permutation matrix $\mathbf{\Pi} \in \mathbb{R}^{n\times n}$. Next lemma gives expressions for weighted and permuted versions of the parameters of A recovered by the algorithm

Lemma 1.3. *The following expressions hold:*

$$\tilde{\boldsymbol{\alpha}}_0^\top = \boldsymbol{\alpha}_0^\top\mathbf{D}_\mathcal{S}\mathbf{D}_\Sigma\mathbf{\Pi} \ , \tag{46}$$

$$\tilde{\boldsymbol{\alpha}}_\infty = \mathbf{\Pi}^\top\mathbf{D}_\Sigma\mathbf{D}_\mathcal{P}\boldsymbol{\alpha}_\infty \ , \tag{47}$$

$$\tilde{\mathbf{T}} = \mathbf{\Pi}^\top\mathbf{D}_\mathcal{S}^{-1}\mathbf{T}\mathbf{D}_\mathcal{S}\mathbf{D}_\Sigma\mathbf{\Pi} \ . \tag{48}$$

Proof. We start by noting the two following identities for the matrices $\tilde{\mathbf{O}}_\mathcal{P} \in \mathbb{R}^{\mathcal{P}\times n}$ and $\tilde{\mathbf{O}}_\mathcal{S} \in \mathbb{R}^{\mathcal{S}\times n}$ used in the algorithm:

$$\tilde{\mathbf{O}}_\mathcal{P} = \bar{\mathbf{H}}_{\mathcal{P},\Sigma}(\tilde{\mathbf{O}}^\top)^+ \tag{49}$$

$$= (\mathbf{O}_\mathcal{P}\mathbf{D}_\mathcal{S}\mathbf{O}^\top)((\mathbf{O}^\top)^+\mathbf{D}_\Sigma\mathbf{\Pi}) \tag{50}$$

$$= \mathbf{O}_\mathcal{P}\mathbf{D}_\mathcal{S}\mathbf{D}_\Sigma\mathbf{\Pi} \ , \tag{51}$$

$$\tilde{\mathbf{O}}_\mathcal{S}^\top = \tilde{\mathbf{O}}^+\bar{\mathbf{H}}_{\Sigma,\mathcal{S}} \tag{52}$$

$$= (\mathbf{\Pi}^\top\mathbf{D}_\Sigma\mathbf{O}^+)(\mathbf{O}\mathbf{D}_\mathcal{P}\mathbf{O}_\mathcal{S}^\top) \tag{53}$$

$$= \mathbf{\Pi}^\top\mathbf{D}_\Sigma\mathbf{D}_\mathcal{P}\mathbf{O}_\mathcal{S}^\top \ . \tag{54}$$

Now, since $\mathbf{e}_\lambda^\top\mathbf{O}_\mathcal{P} = \boldsymbol{\alpha}_0^\top$, we get $\tilde{\boldsymbol{\alpha}}_0^\top = \mathbf{e}_\lambda^\top\tilde{\mathbf{O}}_\mathcal{P} = \boldsymbol{\alpha}_0^\top\mathbf{D}_\mathcal{S}\mathbf{D}_\Sigma\mathbf{\Pi}$. Similarly, $\mathbf{O}_\mathcal{S}^\top\mathbf{e}_\lambda = \boldsymbol{\alpha}_\infty$ yields $\tilde{\boldsymbol{\alpha}}_\infty = \tilde{\mathbf{O}}_\mathcal{S}^\top\mathbf{e}_\lambda = \mathbf{\Pi}^\top\mathbf{D}_\Sigma\mathbf{D}_\mathcal{P}\boldsymbol{\alpha}_\infty$.

Finally, we analyze the expression for $\tilde{\mathbf{T}} \in \mathbb{R}^{n\times n}$ as follows:

$$\tilde{\mathbf{T}} = \tilde{\mathbf{O}}_\mathcal{P}^+\bar{\mathbf{H}}_{\mathcal{P},\mathcal{S}}\mathbf{H}_{\mathcal{P},\mathcal{S}}^+\tilde{\mathbf{O}}_\mathcal{P} \tag{55}$$

$$= (\mathbf{\Pi}^\top\mathbf{D}_\mathcal{S}^{-1}\mathbf{D}_\Sigma^{-1}\mathbf{O}_\mathcal{P}^+)(\mathbf{O}_\mathcal{P}\mathbf{D}_\Sigma\mathbf{T}\mathbf{S})(\mathbf{S}^+\mathbf{O}_\mathcal{P}^+)(\mathbf{O}_\mathcal{P}\mathbf{D}_\mathcal{S}\mathbf{D}_\Sigma\mathbf{\Pi}) \tag{56}$$

$$= \mathbf{\Pi}^\top\mathbf{D}_\mathcal{S}^{-1}\mathbf{T}\mathbf{D}_\mathcal{S}\mathbf{D}_\Sigma\mathbf{\Pi} \ . \tag{57}$$

□

In the last stage of the algorithm the parameters given in the previous lemma are normalized to obtain a proper HMM. These last steps are analyzed in the following result.

Lemma 1.4. *The algorithm returns an HMM* $A^{\mathbf{\Pi}} = \langle \mathbf{\Pi}\boldsymbol{\alpha}_0, \mathbf{\Pi}\boldsymbol{\alpha}_\infty, \mathbf{\Pi}^\top \mathbf{T}\mathbf{\Pi}, \{\mathbf{\Pi}^\top \mathbf{O}_\sigma \mathbf{\Pi}\}\rangle$ *equivalent to* A *modulo a permutation on the states. In particular, we have* $f_A = f_{A^{\mathbf{\Pi}}}$.

Proof. We basically have to show that the normalization steps recover the right parameters up to permutation. We first observe that $\tilde{\mathbf{D}}_\gamma = \mathbf{\Pi}^\top \mathbf{D}_{\mathcal{P}}^{-1} \mathbf{D}_\Sigma^{-1} \mathbf{D}_{\mathcal{S}}^{-1} \mathbf{\Pi}$ and

$$\tilde{\mathbf{D}}_{\mathcal{S}} = \tilde{\mathbf{O}}^\top \mathbf{H}^{\mathrm{p}}_{\mathcal{P},\Sigma}{}^{+} \tilde{\mathbf{O}}_{\mathcal{P}} \tag{58}$$

$$= (\mathbf{\Pi}^\top \mathbf{D}_\Sigma^{-1} \mathbf{O}^\top)(\mathbf{O}^{\top}{}^{+} \mathbf{O}_{\mathcal{P}}^{+})(\mathbf{O}_{\mathcal{P}} \mathbf{D}_\Sigma \mathbf{D}_{\mathcal{S}} \mathbf{\Pi}) \tag{59}$$

$$= \mathbf{\Pi}^\top \mathbf{D}_{\mathcal{S}} \mathbf{\Pi} \ . \tag{60}$$

Therefore, the vector $\boldsymbol{\beta}$ has the following form:

$$\boldsymbol{\beta} = \tilde{\mathbf{D}}_{\mathcal{S}} \tilde{\mathbf{T}}^{+} \tilde{\mathbf{D}}_\gamma \tilde{\boldsymbol{\alpha}}_\infty \tag{61}$$

$$= (\mathbf{\Pi}^\top \mathbf{D}_{\mathcal{S}} \mathbf{\Pi})(\mathbf{\Pi}^\top \mathbf{D}_{\mathcal{S}}^{-1} \mathbf{D}_\Sigma^{-1} \mathbf{T}^{-1} \mathbf{D}_{\mathcal{S}} \mathbf{\Pi})(\mathbf{\Pi}^\top \mathbf{D}_{\mathcal{S}}^{-1} \mathbf{D}_\Sigma^{-1} \mathbf{D}_{\mathcal{P}}^{-1} \mathbf{\Pi})(\mathbf{\Pi}^\top \mathbf{D}_{\mathcal{P}} \mathbf{D}_\Sigma \mathbf{T} \boldsymbol{\alpha}_\infty) \tag{62}$$

$$= \mathbf{\Pi}^\top \mathbf{D}_\Sigma^{-1} \boldsymbol{\alpha}_\infty \ . \tag{63}$$

Recalling that $\mathbf{D}_\Sigma = \mathrm{diag}(\mathbf{1} - \boldsymbol{\alpha}_\infty)$ we see that $\boldsymbol{\beta}(i)/(1 + \boldsymbol{\beta}(i)) = (\mathbf{\Pi}^\top \boldsymbol{\alpha}_\infty)(i)$. Therefore, the final weights returned by the algorithm correspond to $\boldsymbol{\alpha}_\infty^{\mathbf{\Pi}} = \mathbf{\Pi}^\top \boldsymbol{\alpha}_\infty$. Furthermore, we can see that $\tilde{\mathbf{D}}_\Sigma = \mathbf{I} - \mathrm{diag}(\boldsymbol{\alpha}_\infty^{\mathbf{\Pi}}) = \mathbf{\Pi}^\top \mathbf{D}_\Sigma \mathbf{\Pi}$.

Now the recovered weighting matrices can be used to reweight the rest of parameters. Using the expressions from Lemma 1.3 we get

$$\boldsymbol{\alpha}_0^{\mathbf{\Pi}} = \tilde{\mathbf{D}}_{\mathcal{S}}^{+} \tilde{\mathbf{D}}_\Sigma^{+} \tilde{\boldsymbol{\alpha}}_0 = \mathbf{\Pi}^\top \boldsymbol{\alpha}_0 \ , \tag{64}$$

$$\mathbf{T}^{\mathbf{\Pi}} = \tilde{\mathbf{D}}_{\mathcal{S}} \tilde{\mathbf{T}} \tilde{\mathbf{D}}_\Sigma^{+} \tilde{\mathbf{D}}_{\mathcal{S}}^{+} = \mathbf{\Pi}^\top \mathbf{T} \mathbf{\Pi} \ , \tag{65}$$

$$\mathbf{O}^{\mathbf{\Pi}} = \tilde{\mathbf{O}} \tilde{\mathbf{D}}_\Sigma = \mathbf{O}\mathbf{\Pi} \ . \tag{66}$$

Checking that the HMM $A^{\mathbf{\Pi}}$ realizes the same function as A is a rutinary computation. □

2 Implementation Details

This section outlines important implementation details. All methods were implemented using a combination of Python (with the SciPy [Jones et al., 2001] stack) and C++. We note that for all methods separate hyperparameter optimization was performed to optimize for the two contrasting metrics (i.e., WER and perplexity). All code is available at: `https://github.com/ICML14MoMCompare/MoMs-for-StochasticLanguages`.

2.1 The Spectral Method

The spectral method implementation closely mirrors the description in the primary text. However, randomized SVD [Halko et al., 2011] is used in place of an exact truncated SVD, as the randomized method is known to provide robust empirical performance while drastically reducing computational costs [Halko et al., 2011]. In addition, the feature-variances of the estimated Hankel matrices are normalized by independently scaling each row and column by a factor c_u, given by:

$$c_u = \sqrt{\frac{m}{\mathrm{count}(u) + \kappa}}, \tag{67}$$

where u is the prefix/suffix corresponding to the row/column, $\mathrm{count}(u)$ is the number of times that prefix/suffix occurs in the training data, m is the number of strings in the training set, and κ is a smoothing

constant (following Cohen et al. [2013], a value of $\kappa = 5$ was used). This scaling serves to normalize the estimates such that the empirical variances for estimates of frequently encountered prefixes/suffixes and infrequently encountered prefixes/suffixes are roughly equal (see Cohen et al. [2013] for justification of this technique in the context of spectral learning).

For the spectral methods, the model-size hyper-parameter selection is performed via a two-stage grid search. In the first stage, models of sizes 10 through 70 (inclusive) are examined (incrementing by 10). Following this, the grid search is performed over sizes in the range [n-9,n+9], where n is the best size determined from the first phase.

2.1.1 Spec-Str

For the Spec-Str method, where estimates $\hat{f}_S$ are used, we set $|\mathcal{P}| = |\mathcal{S}| = 10000$, including the top $k = 10000$ prefixes/suffixes according to their empirical frequency. Using such large bases requires operations on very large sparse matrices; thus numerous sparse representations provided by the SciPy and Eigen [Guennebaud et al., 2010] libraries are exploited in this method.

2.1.2 Spec-Sub

For the Spec-Sub method, where estimates $\hat{f}^{\mathrm{s}}_S$ are used, we set $|\mathcal{P}| = |\mathcal{S}| = 500$, as this provided accurate results with minimal computational cost. Again the top $k = 500$ prefixes/suffixes according to empirical frequency were selected; however, in this case the max length of a substring prefix/suffix was set to four (shorter sequences are more frequent so this choice did not have significant impact on the selected prefixes/suffixes). We note that in this setting the Hankel estimates are dense; thus, computations involving the matrices are considerably more expensive (justifying the smaller choice of basis compared to Spec-Str).

2.2 Convex Optimization

The convex optimization method required significant implementation work compared to the other methods. In this work, we chose to use the alternating direction method of multipliers (ADMM) [Boyd et al., 2011] as the convex optimization routine, as this approach has desirable speed and convergence properties. Details of this convex optimization routine can be found in Boyd et al. [2011].

However, despite the efficiency of ADMM, the CO moment-based algorithm is quite slow without additional implementation tweaks, as the optimization requires an SVD of the Hankel matrix to be performed at each iteration. In order to increase computational efficiency, we implemented an over-relaxed version of ADMM with inexact proximal operators, where a randomized truncated SVD is used instead of a full SVD. During these truncated steps, only the top 50 singular values are used. This version essentially uses an (more efficient) approximate proximal gradient operator during the ADMM updates, a technique which still guarantees convergence under some conditions (see Boyd et al. [2011] for details). After running a number of iterations of this modified ADMM (stopping either via convergence or an iteration limit), we then applied four iterations using a full SVD.

The hyper-parameter optimization for the τ regularization constant was performed analogously to the spectral methods, with values in the range $[10^{-5}, \zeta]$ examined, with

$$\zeta = \frac{\sqrt{|\mathcal{P}||\mathcal{S}|}\sigma_1^{\mathcal{H}}}{\sqrt{m}}, \tag{68}$$

where $\sigma_1^{\mathcal{H}}$ is the largest singular value of the empirical Hankel matrix and m is the number of strings in the training set. See Balle [2013] for justification of this upper-bound.

We use the top-k frequent sequences with $|\mathcal{P}| = |\mathcal{S}| = 50$ in this work for optimization for perplexity and $|\mathcal{P}| = |\mathcal{S}| = 200$ for WER, as preliminary experiments demonstrated that a smaller basis was advantageous for accuracy according to perplexity and the larger basis for WER. We note that bases of sizes larger than 200 become prohibitively expensive, given the computational requirements of the ADMM optimization. Moreover, the final learned representation of the CO method scales linearly with the size of the bases used, so

using large bases introduces significant overhead. As with Spec-Sub, estimates $\hat{f}^{\mathrm{s}}_S$ are used and prefix/suffix sequences are limited to a max-length of four.

2.3 Tensor Decomposition

The implementation of the tensor method largely follows from the description in the primary text. As stated in the primary text, the tensor power method of Anandkumar et al. [2013] is used as the decomposition method. The maximum number of tensor power iterations was set to 100. In contrast to the description in Anandkumar et al. [2013], we did not use random restarts during the power iterations, as this was not observed to increase the quality of the solutions and introduces extra computational costs.

Moreover, in order to make the algorithm scalable it is necessary to exploit the extreme sparsity of the estimated Hankel matrices. The sparse matrix representations used are identical to those used with the Spec-Str method. In addition to these sparse representations, we utilize the sparse LSQR method [Paige and Saunders, 1982] for solving least-squares. This method is used to implicitly compute the pseudoinverse, $\mathbf{H}^{+}_{\mathcal{P},\mathcal{S}}$, as explicitly representing $\mathbf{H}^{+}_{\mathcal{P},\mathcal{S}}$ is prohibitively expensive given that the pseudoinverses of sparse matrices are, in general, not sparse.

We use a sparse SVD via the Lanczos algorithm (provided by SciPy) to compute the $\mathbf{Q}_{\mathcal{P}}$ and $\mathbf{Q}_{\mathcal{S}}$ matrices. Random projections could also be used, but we found the SVD generated matrices to give superior performance.

As with the Spec-Str method, a basis size of $|\mathcal{P}| = |\mathcal{S}| = 10000$ is used (with the most empirically frequent sequences selected). For this method, one grid search is used to determine the optimal model size, as the space of possible model sizes is restricted to $[1, |\Sigma|]$.

2.4 EM

The Treba library [Hulden, 2012] with the default convergence criterion was used in conjunction with a hand-coded hyper-parameter optimization scheme (Treba does not provide such optimization). For hyper-parameter optimization, model sizes in the set {5,10,20,30,40} are examined. For each model-size, we first run five random-restarts for N_1/n (where n is the model-size and N_1 a constant) iterations, selecting the best-performing model for further optimization. Following this, optimization is interleaved with validation to check for overfitting. Specifically, every N_1/n iterations, the model is scored. If the model is not improving for three such checks then training is terminated, and the top-performing model among all validation checks is returned. In addition, a maximum of N_2/n such validation-checks are performed. We set $N_1 = 100$ and $N_2 = 1000$, as this led to reasonable runtimes (and the routine rarely reached the maximum iteration limit). The number of iterations (both between validations and in total) being inversely proportional to model size is a pragmatic specification, as the larger models were significantly (several orders-of-magnitude) slower to train.

2.5 EM-Tensor

For EM initialized with the solution of the tensor method, we simply interleave five iterations of Baum-Welch EM optimization with validation checks and stop training when there is no improvement for three validation-checks (the top-performing models being successively stored).

References

A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *CoRR*, abs/1210.7559, 2012.

A. Anandkumar, R. Ge, D. Hsu, and S. Kakade. A tensor spectral approach to learning mixed membership community models. In *COLT*, 2013.

B. Balle. *Learning Finite-State Machines: Algorithmic and Statistical Aspects*. PhD thesis, Universitat Politècnica de Catalunya, 2013.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 2011.

S. B. Cohen, K. Stratos, M. Collins, D. P. Foster, and L. Ungar. Experiments with spectral learning of latent-variable PCFGs. In *NAACL-HLT*, 2013.

Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

N. Halko, P. Martinsson, and J. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

M. Hulden. Treba: Efficient numerically stable EM for PFA. In *ICGI*, 2012.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

C. Paige and M. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.