Bayesian Learning of Non-Negative Matrix/Tensor Factorizations by Simulating Pólya Urns

M. Burak Kurutmaz[1], A. Taylan Cemgil[1], Umut Şimşekli[2], Melih Barsbey[1], Sinan Yıldırım[3]
[1]Boğaziçi University, [2]Télécom ParisTech, [3]Sabancı University

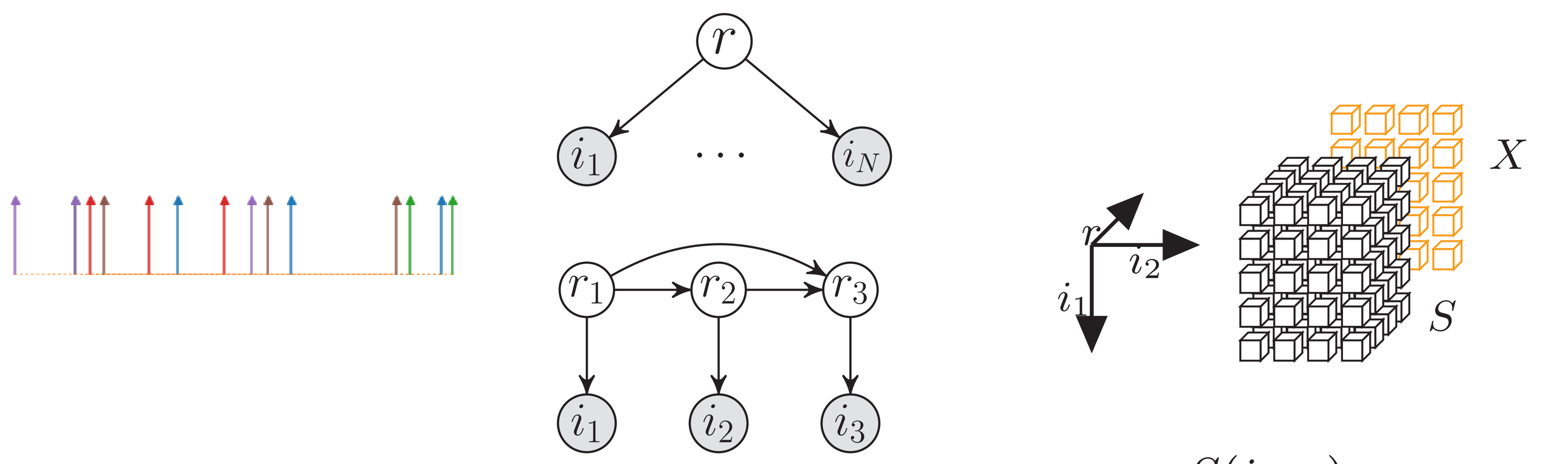
Introduction & Motivation

- **Tensor Decompositions:**
$$X(i_V) \approx \hat{X}(i_V) = \sum_{i_{\bar{V}}} \prod_{\alpha} \theta_\alpha(i_\alpha), \qquad \Theta^* = \arg\min KL(X\|\hat{X})$$
- **Example** (PARAFAC): $X(i_1, \cdots, i_N) \approx \sum_r \prod_{n=1}^N \theta_n(r, i_n)$
- We propose a **Poisson Process** formulation unifying **KL-NTF**, **Topic Models** and **Discrete Bayesian Networks** which naturally leads to a **SMC** algorithm for marginal likelihood: $p(X) = \int d\Theta p(X|\Theta)p(\Theta)$
- The **computational complexity** of the algorithm scales with the **total sum** of the elements in X, and does not depend on the **size** of X.
- We illustrate with examples that our algorithm gives promising results as a practical algorithm in the **sparse data regime**.

Bayesian Allocation Model

- **Bayesian Allocation Model** (BAM): joint model of counts thinned from a **Poisson process** by a **Bayesian Network** $\mathcal{G}$: Tokens allocated to a **tensor** S:

$s^t \sim \mathcal{PP}(\lambda)$ — Thinning $\mathcal{G}$ — Allocation Tensor $S(i_{1:N})$



$\lambda \sim \mathcal{GA}(a, b)$ — $\theta_n \sim \mathcal{D}(\alpha_n)$ — $S(i_{1:N}) \sim \mathcal{PO}\left(\lambda \prod_n \theta_n(i_n, i_{\mathrm{pa}(n)})\right)$

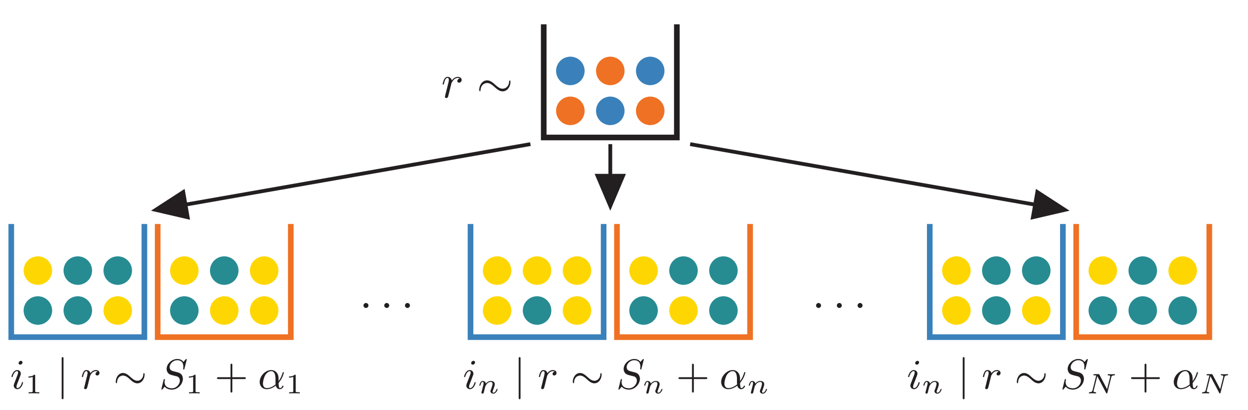
- Allocation tensor is **observed** only in dimensions $V \subset [N]$:
$$X(i_V) = \sum_{i_{\bar{V}}} S(i_V, i_{\bar{V}})$$

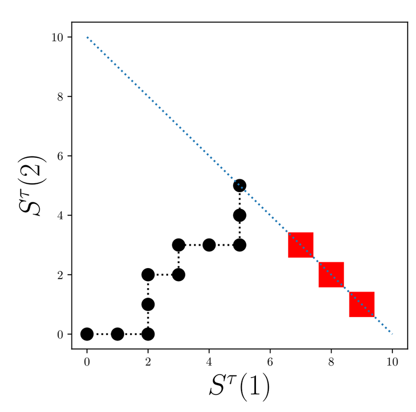
Marginal Distribution of Allocation Tensor

- By analytically integrating out the parameters:
$$p(S_+|a,b) = \frac{\Gamma(\alpha_+ + S_+)}{\Gamma(a)\Gamma(S_+ + 1)} \left(\frac{b}{b+1}\right)^a \left(\frac{1}{b+1}\right)^{S_+}$$
$$p(S|S_+, \alpha) = \left(\prod_n \frac{B_n(\alpha_n + S_n)}{B_n(\alpha_n)}\right) \frac{1}{B(S+1)}$$

where $B_n(Z_n) = \prod_{i_n} \Gamma(Z(i_n, i_{\mathrm{pa}(n)}))/\Gamma(\sum_{i_n} Z(i_n, i_{\mathrm{pa}(n)}))$.

Marginal BAM as a Pólya Urn Process





- **Forward Process**: Replace Dirichlet θ_n's with **Pólya Urns**
$$p(s^\tau(i_{1:N}) = 1 \mid S^{\tau-1}) = \prod_{n=1}^N \frac{\alpha_n(i_n, i_{\mathrm{pa}(n)}) + S_n^{\tau-1}(i_n, i_{\mathrm{pa}(n)})}{\sum_{i_n'} \alpha_n(i_n', i_{\mathrm{pa}(n)}) + S_n^{\tau-1}(i_n', i_{\mathrm{pa}(n)})}$$
- **Backward Process**: Sampling **without** replacement
$$p(s^\tau(i_{1:N}) = 1 \mid S^\tau) = \prod_{n=1}^N \frac{S_n^\tau(i_n, i_{\mathrm{pa}(n)})}{\sum_{i_n'} S_n^\tau(i_n, i_{\mathrm{pa}(n)})}$$
$$s_V^{\tau+1} \mid X, S_V^\tau \sim L(s_V^{\tau+1}|X, S^\tau) = \prod_{i_V} \left(\frac{X(i_V) - S_V^\tau(i_V)}{T - \tau}\right)^{s_V^{\tau+1}(i_V)}$$

Evidence Lower Bound

- **Mean-field factorization** assumption yields with
$$q(\theta_n) = \mathcal{D}(\beta_n) \qquad q(\lambda) = \mathcal{G}(c, d) \qquad q(S(:, i_V)) = \mathcal{M}(X(i_V), \pi(:| i_V))$$
where π, c, d and β_n's are the the variational parameters.
- The **evidence lower bound** (ELBO) is given as
$$e^{\mathcal{B}} = \frac{b^a}{d^c} \frac{\Gamma(c)}{\Gamma(a)} \left(\prod_n \frac{B_n(\beta_n)}{B_n(\alpha_n)}\right) \frac{\prod_{i_{1:N}} \pi(i_{\bar{V}}, i_V)^{-\langle S(i_{1:N})|X\rangle}}{\prod_{i_V} \Gamma(X(i_V) + 1)}$$

Sequential Monte Carlo

- Choose $q(s^{1:T}) = \prod_\tau L(s_V^\tau|X, S^{\tau-1}) p(s_{\bar{V}}^\tau|s_V^\tau, S^{\tau-1})$

procedure BAM-SIS(X)
$Z^0 = 1$
$S^0 = 0$
$c = \frac{b^a}{(b+1)^{a+S_+}} \frac{\Gamma(a+S_+)}{\Gamma(a)} \frac{1}{S_+!}$
for $\tau = 1, \ldots, T = X_+$ **do**
  Sample $s_V^\tau \sim L(s_V^\tau|X, S^{\tau-1})$
  Sample $s_{\bar{V}}^\tau \sim p(s_{\bar{V}}^\tau|s_V^\tau, S^{\tau-1})$
  $s^\tau = s_V^\tau \otimes s_{\bar{V}}^\tau$
  $S^\tau = S^{\tau-1} + s^\tau$
  $u^\tau = p(s_V^\tau|S^{\tau-1}) \prod_{i_V} \left(S_{V+}^\tau / S_V^\tau(i_V)\right)^{s_V^\tau(i_V)}$
  $Z^\tau = Z^{\tau-1} u^\tau$
return $c \times Z^T, S$

Experiments

- **Toy Example:** $X = \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{pmatrix}$

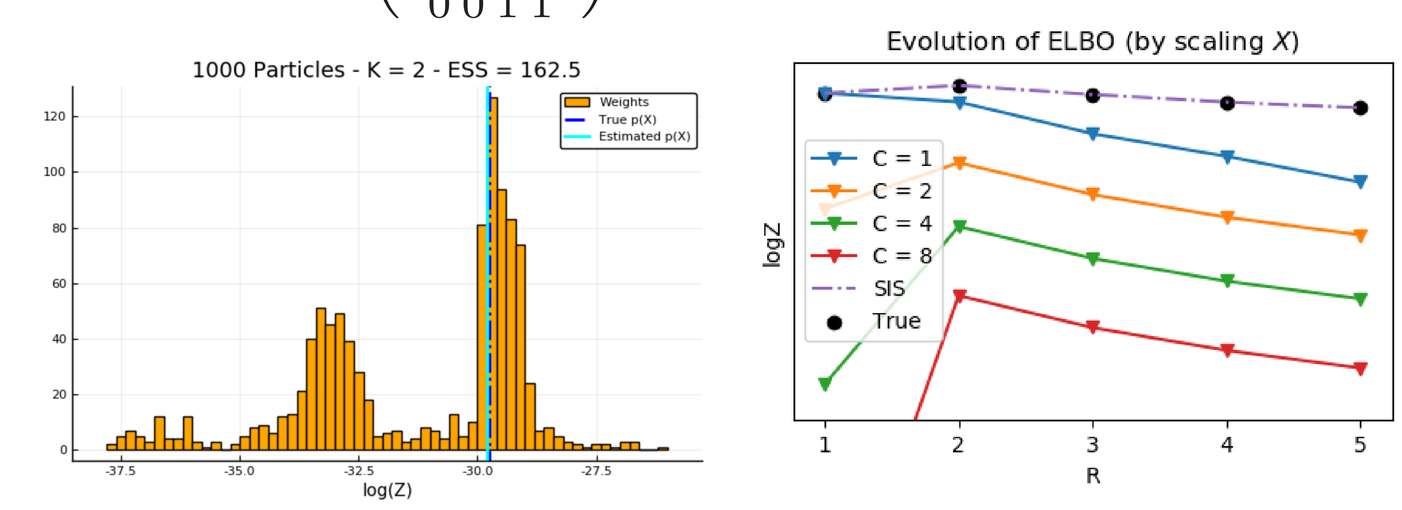


Figure 1: (a) Histogram of particle weights. (b) SMC estimates true $p(X)$ hence the model order accurately, while ELBO can do so only at denser data regimes.

- **Synthetic Data Experiments**

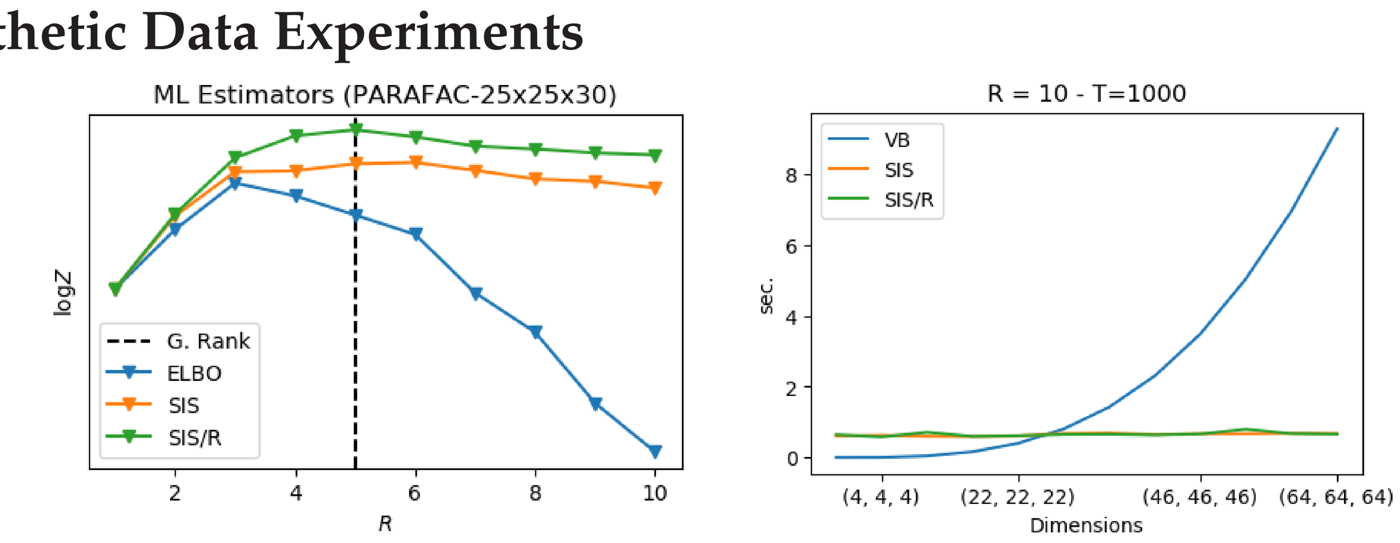


Figure 2: (a) SMC with adaptive particle filtering correctly detects the generated rank of the tensor. (b) Runtime of mean field variational inference scales with the size of X; whereas the runtime of SMC algorithms are independent of it.

- **Modeling letter transitions** (`norvig.com/mayzner.html`)
$$\underbrace{X(i,j) \approx \sum_r \theta_0(r)\theta_1(i,r)\theta_2(j,r)}_{\text{NMF}\equiv\text{LDA}} \text{ vs. } \underbrace{X(i,j) \approx \sum_r \theta_0(r)\theta_1(i,r)\theta_1(j,r)}_{\text{symmetric NMF}}$$

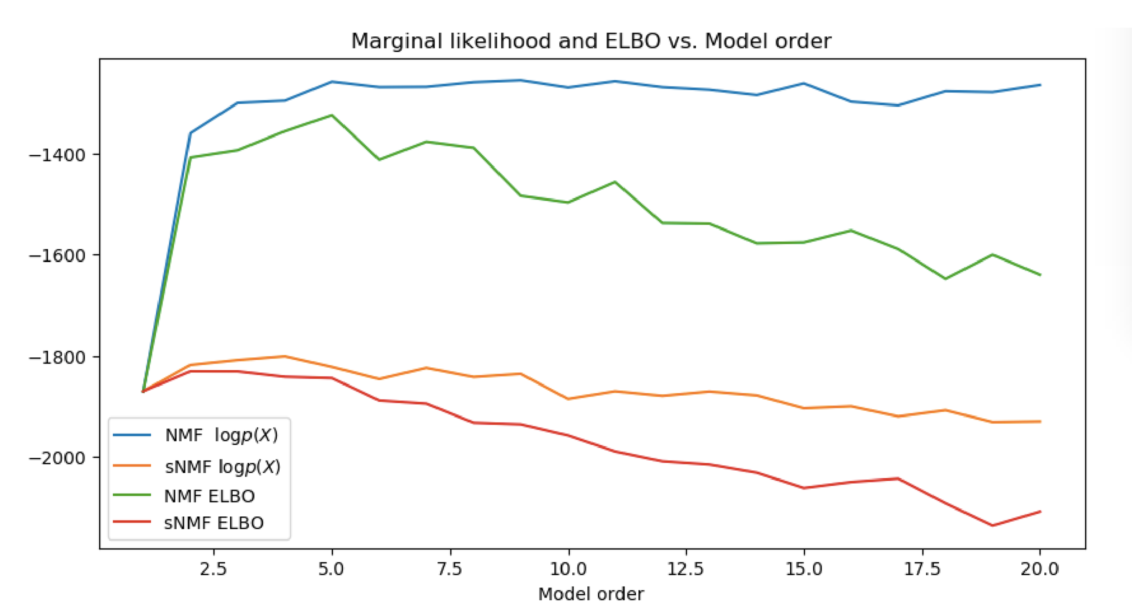


Figure 3: Comparison of NMF and sNMF models: As expected model likelihood (and ELBO) for the NMF is higher than sNMF for all model orders.

Conclusion

- Bayesian Allocation Model, highlights the connections of nonnegative tensor factorizations and discrete Bayes networks.
- A justification of VB in dense data regime by bringing an alternative perspective to sample size.
- A direct explanation of the "by parts representation" nature of NMF/NTF via self-reinforcing Pólya urns.
- It offers a natural SMC algorithm that scales with sum but not the observed size of the tensor, hence practical for sparse tensors with large dimensions and provides a model scoring method.