Streaming Inference for Infinite Feature Models

Rylan Schaeffer [1, 2] Yilun Du [3] Gabrielle Kaili-May Liu [2] Ila Rani Fiete [2, 4]

[1]Stanford Computer Science [2]MIT Brain and Cognitive Sciences [3]MIT EECS [4]McGovern Institute for Brain Research

Summary

- Biological intelligence must contend with unsupervised, streaming data. How should one approach machine learning in this data regime?
- We consider **feature models**, a class of unsupervised models that attempts unsupervised discovery of latent features underlying data and that encompasses PCA, FA, ICA & NMF
- We make feature models significantly more applicable to streaming data by imbuing them with the ability to create new features, online, in a probabilistic and principled manner
- To achieve this, we derive a novel recursive form of the Indian Buffet Process (IBP), which we term the **Recursive IBP** (R-IBP)
- We show on synthetic and real data that R-IBP achieves comparable or better performance in significantly less time than existing sampling and variational baselines

Notation and Background

- **Observations:** $o_{1:N}$ where $o_n \in \mathbb{R}^D$
- **Features:** $\{A_k\}_{k=1}^K$ where $A_k \in \mathbb{R}^D$ and K unknown
- **Indicators:** $z_{1:N}$ where $z_n \in \{0,1\}^K$ and K unknown
- **Generative model:**

$$\begin{aligned} z_{1:N} &\sim IBP(\alpha, \beta) \\ \{A_k\} &\sim p(\{A_k\}) \\ o_n | z_n, \{A_k\} &\sim p(o | z_n, \{A_k\}) \end{aligned}$$

- **Indian Buffet Process (IBP) [1]:** The IBP is a 2-parameter $\alpha > 0, \beta > 0$ stochastic process defining a distribution over binary matrices with finitely many rows and unbounded number of columns. Let $\lambda_n \sim Poisson(\alpha\beta/(\beta + n - 1))$ and $\Lambda_n \overset{\text{def}}{=} \sum_{n'=1}^{n'=n} \lambda_{n'}$. Then $IBP(\alpha, \beta)$ is:

$$p(z_{nk} = 1 | z_{<n,k}, \Lambda_{n-1}, \lambda_n, \alpha, \beta) \overset{\text{def}}{=} \begin{cases} \frac{1}{\beta+n-1} \sum_{n'<n} z_{n'k} & \text{if } 1 \leq k \leq \Lambda_{n-1} \\ 1 & \text{if } \Lambda_{n-1} < k \leq \Lambda_{n-1} + \lambda_n \\ 0 & \text{otherwise} \end{cases}$$

Goal: Streaming Inference for Infinite Feature Models

Filter a posterior over the current observation's binary latent variables $z_n \overset{\text{def}}{=} \{z_{nk}\}_{k=1}^{k=\infty}$ and the latent features $\{A_k\}_{k=1}^{\infty}$, given the entire history of observations $o_{\leq n}$, subject to two constraints: (1) Inference must be performed online, i.e., the nth observation is discarded before proceeding, (2) Inference must be efficient in the large N limit

Challenges

1. **Dependence on Entire History**: The IBP's conditional distribution $p(z_{nk} | z_{<nk}, \Lambda_{n-1}, \lambda_n)$ renders the current indicators z_n dependent on *all* previous indicators $z_{<n}$
2. **Exponentially Many Evaluations of Likelihood**: z_n is the set of binary variables $\{z_{nk}\}_{k=1}^{k=\Lambda_n}$, meaning the likelihood must be evaluated for 2^{Λ_n} possible configurations at each step
3. **Non-Factorized Posterior**: In the prior, the indicators are independent, i.e., $p(z_n | z_{<n}, \Lambda_{n-1}, \lambda_n) = \prod_{k=1}^{k=\Lambda_n} p(z_{nk} | z_{<nk}, \Lambda_{n-1}, \lambda_n)$. After conditioning on observations, the indicators are no longer independent, i.e., $p(z_n | o_{\leq n}) \neq \prod_{k=1}^{k=\Lambda_n} p(z_{nk} | o_{\leq n})$
4. **Unknown Posterior over Number of Features**: What are the posteriors for the number of new features λ_n and the total number of features Λ_n?

The Recursive Indian Buffet Process (R-IBP)

Idea: Break the dependence on the entire history by converting the IBP's conditional distribution $p(z_{nk} = 1 | z_{<n}, \Lambda_{n-1}, \lambda_n, \alpha, \beta)$ into a sequence of marginal distributions $p(z_{nk} = 1 | \alpha, \beta)$ that can be efficiently computed recursively, similar to [2]:

$$p(z_{nk} = 1 | \alpha, \beta) = \frac{1}{\beta + n - 1} \sum_{n'<n} p(z_{n'k} = 1) + p(\Lambda_{n-1} \leq k - 1) - p(\Lambda_{n-1} + \lambda_n \leq k - 1) \qquad (1)$$

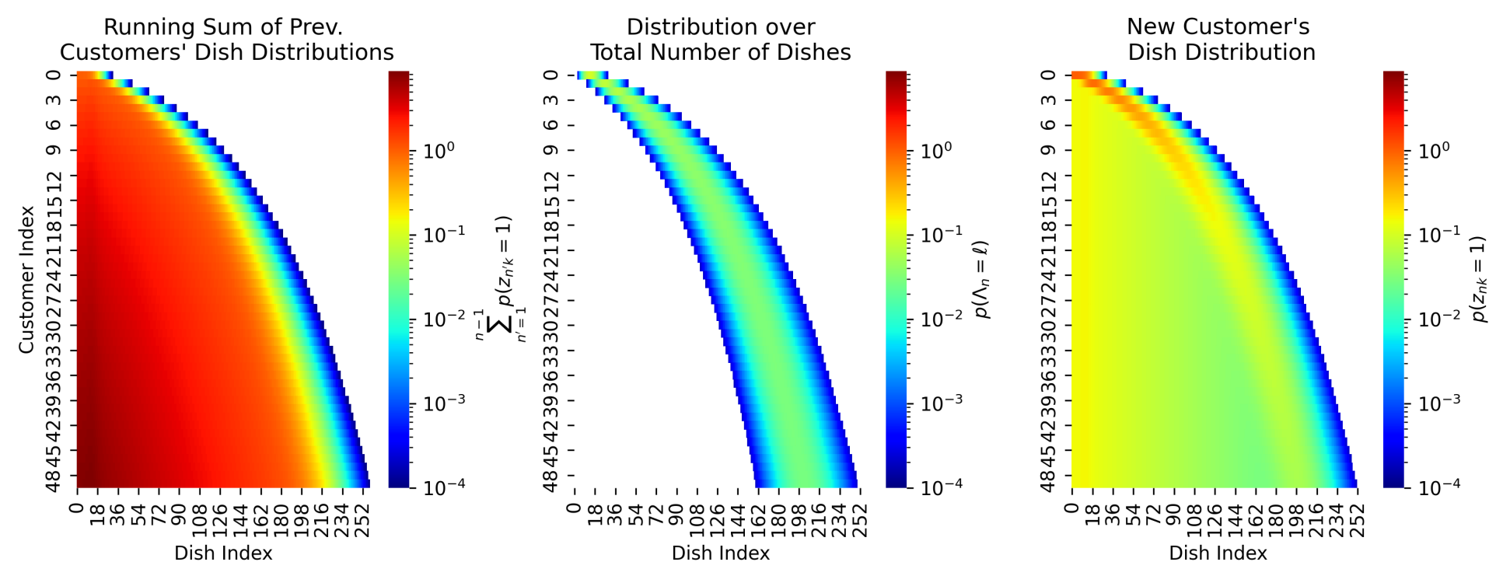

Figure 1. **Visualization of the Recursive IBP**. Intuitively, the probability that the kth feature is present in the nth observation is given by the running sum of how probable the kth feature's presence was in all previous observation (left), plus the difference of two Poisson CDFs that drives new observations to create new features (center).

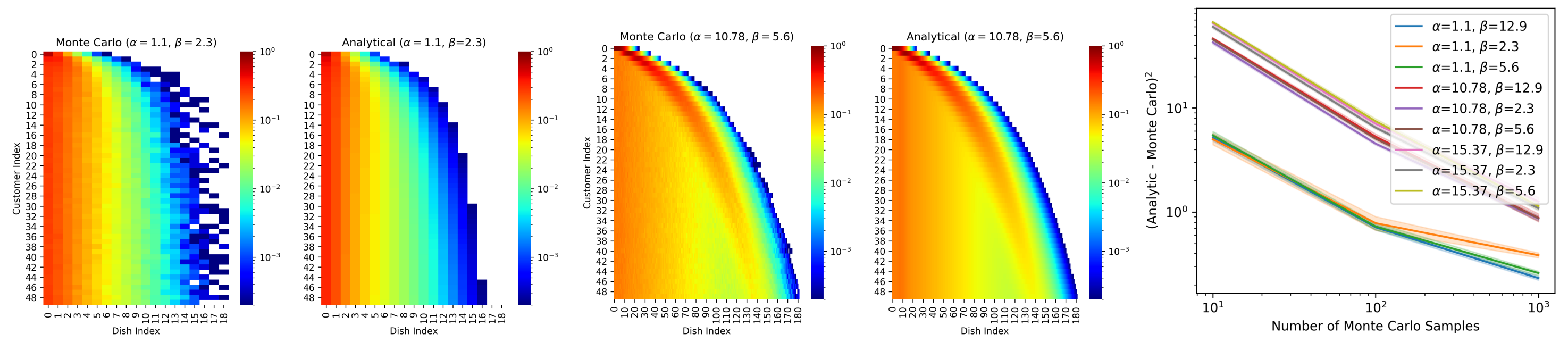

Figure 2. **(Left) Monte Carlo vs. Analytical Expression.** Over a wide range of (α, β) pairs, we find excellent match between Monte Carlo estimates of the marginal probabilities drawn from the conditional $p(z_n | z_{<n}, \alpha, \beta)$ and the R-IBP's marginal probabilities $p(z_n | \alpha, \beta)$. **(Right) Mean-Squared Error between analytical expression for the marginal and a Monte Carlo marginal estimate.** The mean-squared error falls approximately as a power law.

Analytical Results: R-IBP in the Zero-Noise Limit

Does inference with R-IBP converge? To what? How quickly? Consider a linear-Gaussian model $O = ZA + E$. In the limit $\sigma_o^2 \rightarrow 0$, R-IBP fits the data by minimizing the objective function:

$$\mathcal{L}(Z, A, \Lambda_N) \overset{\text{def}}{=} \Big[(O - ZA)^T (O - ZA)\Big] + \gamma^2 \Lambda_N \qquad (2)$$

Intuition: R-IBP minimizes the squared error between the observations and the subset of infinite features thought to be present, while regularizing the number of features, akin to BIC [3].

Empirical Results: Synthetic Data & MNIST

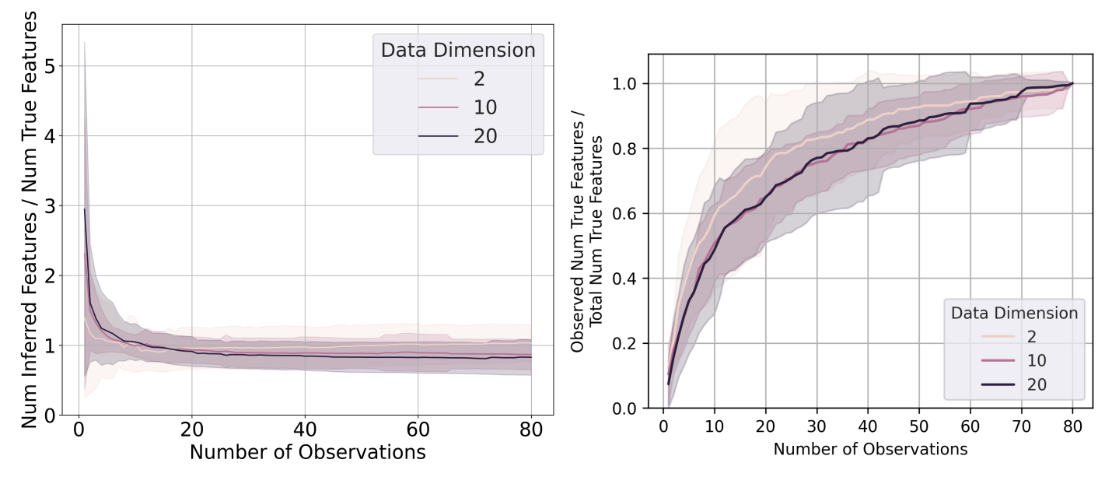

Figure 3. **R-IBP Feature Recovery on Streaming Data.** R-IBP recovers the correct order of magnitude of number of features (left), adding features as more observations are encountered (right).

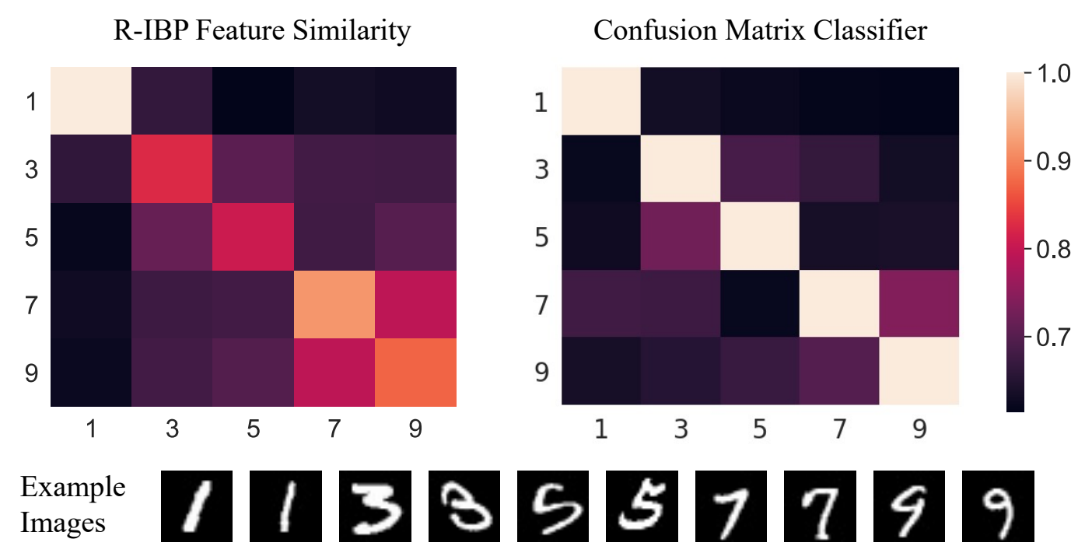

Figure 4. **R-IBP Recovers Intuitive Features for MNIST Classes.** Feature similarity matches the confusion matrix of an independently-trained convolutional neural network classifier.

Empirical Results: Synthetic Data

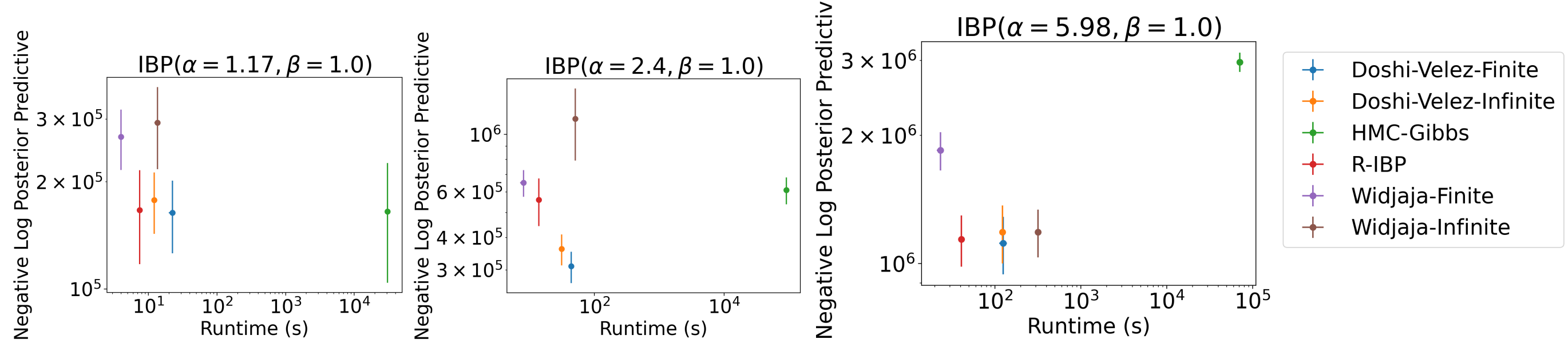

Figure 5. **Comparison of Linear-Gaussian Inference Algorithms.** Over a range of α values, R-IBP is significantly faster than baseline inference algorithms and has better (lower) negative log posterior predictive values than the streaming baselines and even some non-streaming baselines, averaged over 10 synthetic datasets. We fix $\beta = 1.0$ because baseline algorithms are only defined for $\beta = 1.0$. The correct α, β values are assumed known.

Empirical Results: UCI Tabular Data
2014 Diabetic Patients & 2016 Cancer Gene Expression

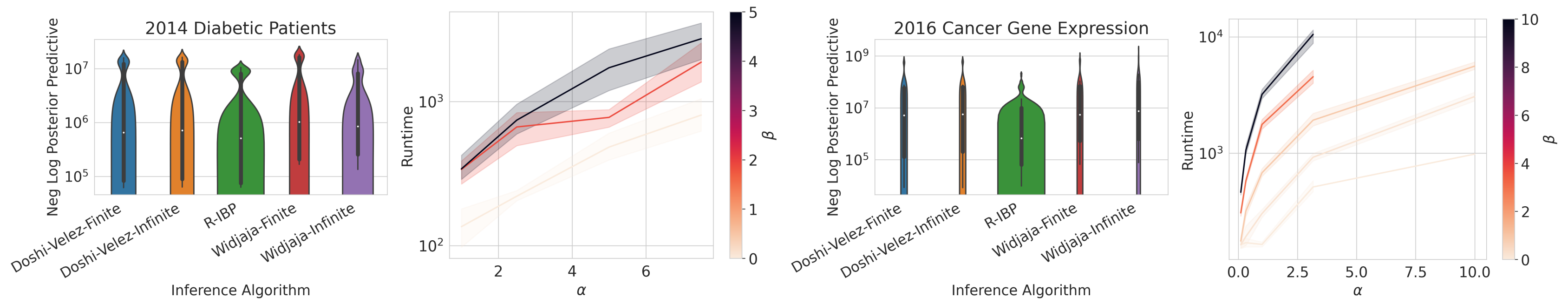

Figure 6. **R-IBP performance on diabetic patient data and cancer gene expression.** R-IBP matches or outperforms baseline algorithms across hyperparameter configurations. R-IBP runtime scales linearly with α and quasilinearly with β (right), qualitatively matching our complexity analysis.

Conjecture: Graphical Structure Prevents Multiplicative Errors

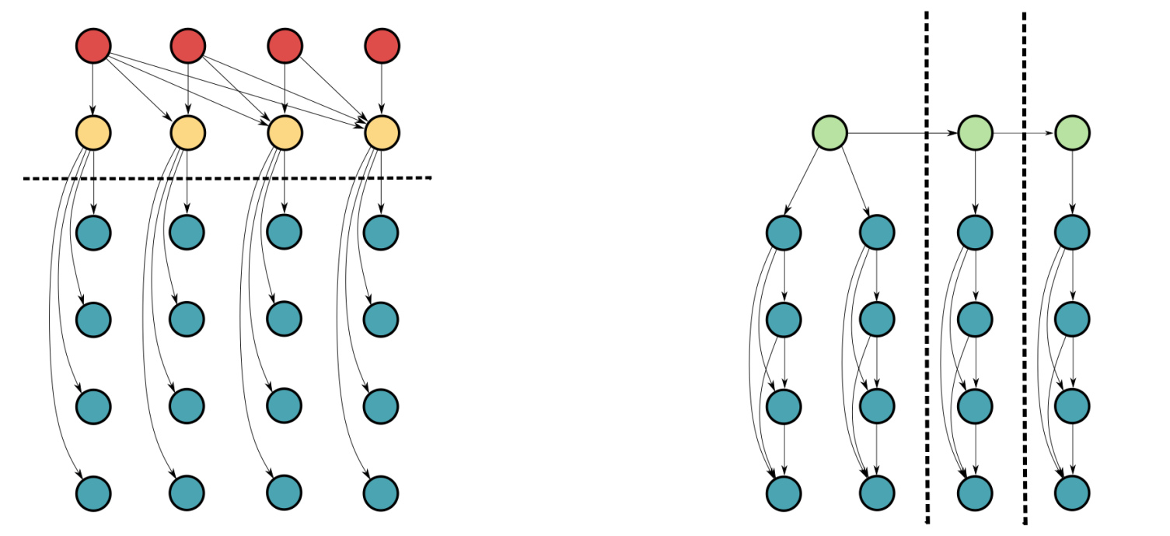

Figure 7. **Beta Process (BP) vs. R-IBP.** Many baseline algorithms are based on the BP (left), which chain multiplies terms (red) to compute each feature's probability (yellow) for Z (aqua). In contrast, the Recursive IBP (right) creates columns (green), then adds terms within columns of Z (aqua). We conjecture R-IBP's adding inferred quantities to running sums, rather than chain multiplying inferred quantities, prevents errors from compounding and enables R-IBP to outperform even non-streaming baseline algorithms based on the Beta Process.

References

[1] Griffiths and Ghahramani. *NeurIPS*, 2005.

[2] Schaeffer and et al. *Uncertainty in Artificial Intelligence*, 2021.

[3] Schwarz. *The Annals of Statistics*, 1978.

rylanschaeffer.github.io

ICML 2022

rylanschaeffer@gmail.com