Improvements to the Sequence Memoizer

Jan Gasthaus
Yee Whye Teh

Presented by
Zachary Kahn

Overview

- Sequence Memoizer(SM) is based on Hierarchal Pitman-Yor Process(HPYP) with free parameters
  - SM as seen before sets these to 0, new model allows flexibility
- SM algorithms use Chinese Restaurant Franchise(CRF) representations for HPYP
  - Needs to store lists of customers at each table, lots of memory

Pitman-Yor Process Notation

- $PY(\alpha,d,G_0)$=Pitman-Yor Process
  - Concentration parameter $\alpha>-d$, discount parameter $d\in[0,1)$, Base distribution G_0 over probability space Σ
- C customers indexed as $[c]=\{1,\ldots,c\}$
  - Seating arrangements are sets of disjoint non-empty subset partitioning of [c], e.g. $\{\{1,3\},\{2\}\}$
- A_c = set of seating arrangements for c customers, A_{ct} subset with exactly t tables

Probability Distributions

- New customers join table a with probability $\frac{|a|-d}{a+c}$ and start new table with probability $\frac{\alpha+|A|d}{a+c}$

$$P(A) = \frac{[\alpha + d]_d^{|A|-1}}{[\alpha + 1]_1^{c-1}} \prod_{a \in A} [1 - d]_1^{|a|-1} \quad \text{for each } A \in \mathcal{A}_c,$$

$$[y]_d^n = \prod_{i=0}^{n-1} y + id$$

- Fixing t≤c

$$P(A) = \frac{\prod_{a \in A} [1 - d]_1^{|a|-1}}{S_d(c,t)} \quad S_d(c,t) = \sum_{A \in \mathcal{A}_{ct}} \prod_{a \in A} [1 - d]_1^{|a|-1}$$

Inference

$$P(\{c_s, t_s, A_s\}, z_{1:c}) = \left(\prod_{s\in\Sigma} G_0(s)^{t_s}\right)\left(\frac{[\alpha+d]_d^{t_\cdot-1}}{[\alpha+1]_1^{c_\cdot-1}}\prod_{s\in\Sigma}\prod_{a\in A_s}[1-d]_1^{|a|-1}\right), \tag{3}$$

$$P(\{c_s, t_s\}, z_{1:c}) = \left(\prod_{s\in\Sigma} G_0(s)^{t_s}\right)\left(\frac{[\alpha+d]_d^{t_\cdot-1}}{[\alpha+1]_1^{c_\cdot-1}}\prod_{s\in\Sigma} S_d(c_s, t_s)\right). \tag{4}$$

- $G \sim PY(\alpha, d, G_0)$ $z_1, \ldots, z_n | G \sim G$
- z_i=dish served at customer i's table
- $s \in \Sigma$ = a dish
- c_s=number z_i served dish s
- t_s=Number of tables served dish s.

Sequence Memoizer

- Σ =Set of symbols to model, Σ^* = Set of finite sequences from Σ
- $G_u(s)$=conditional probability of symbol s after context u.
- ε=empty string, sequence dropping first character in u

$$P(x_{1:T}) = \prod_{i=1}^{T} P(x_i|x_{1:i-1}) = \prod_{i=1}^{T} G_{x_{1:i-1}}(x_i), \quad (5)$$

$$G_\varepsilon \sim \mathrm{PY}(\alpha_\varepsilon, d_\varepsilon, H)$$

$$G_{\mathbf{u}}|G_{\sigma(\mathbf{u})} \sim \mathrm{PY}(\alpha_{\mathbf{u}}, d_{\mathbf{u}}, G_{\sigma(\mathbf{u})}) \qquad \text{for } \mathbf{u} \in \Sigma^*\backslash\{\varepsilon\},$$

Chinese Restaurant Franchise

- The hierarchy over $\{G_u\}$ is represented with a CRF with each G_u is a restaurant indexed by u
  - Customers are draws from G_u, tables drawn from $G_{\sigma(u)}$ and dishes drawn from Σ
  - c_{us} and t_{us} are number of customers and tables in restaurant u served dish s with seating arrangement A_{us}

$$c_{\mathbf{u}s} = c^x_{\mathbf{u}s} + \sum_{\mathbf{v}:\sigma(\mathbf{v})=\mathbf{u}} t_{\mathbf{v}s},$$

$c^x_{\mathbf{u}s} = 1$ if $s = x_i$ and $\mathbf{u} = x_{1:i-1}$ for some i, and 0 otherwise.

CRF Probabilities

$$P(\{c_{\mathbf{u}s}, t_{\mathbf{u}s}, A_{\mathbf{u}s}\}, x_{1:T}) = \left(\prod_{s\in\Sigma} H(s)^{t_{\varepsilon s}}\right) \prod_{\mathbf{u}\in\Sigma^*} \left(\frac{[\alpha_{\mathbf{u}} + d_{\mathbf{u}}]_{d_{\mathbf{u}}}^{t_{\mathbf{u}\cdot}-1}}{[\alpha_{\mathbf{u}} + 1]_1^{c_{\mathbf{u}\cdot}-1}} \prod_{s\in\Sigma} \prod_{a\in A_{\mathbf{u}s}} [1 - d_{\mathbf{u}}]_1^{|a|-1}\right). \tag{8}$$

$$P^*_{\mathbf{v}}(s) = \frac{c_{\mathbf{v}s} - t_{\mathbf{v}s} d}{\alpha_{\mathbf{v}} + c_{\mathbf{v}\cdot}} + \frac{\alpha_{\mathbf{v}} + t_{\mathbf{v}\cdot} d}{\alpha_{\mathbf{v}} + c_{\mathbf{v}\cdot}} P^*_{\sigma(\mathbf{v})}(s). \tag{9}$$

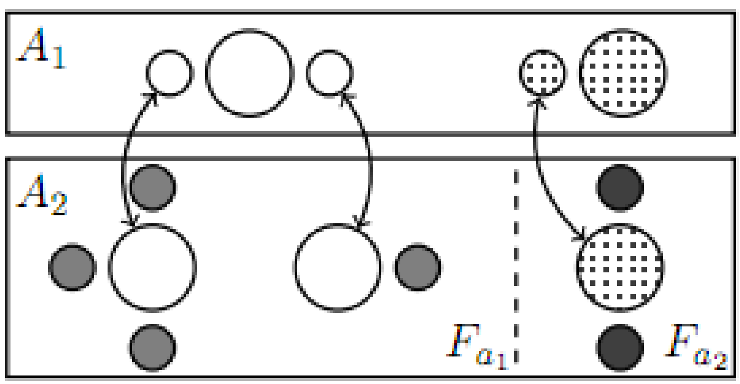
Nonzero Concentrations

- Previous models set $\alpha=0$
- We set $\alpha_\varepsilon = \alpha > 0$, $\alpha_u = \alpha_{\sigma(u)} d_u > 0$
- This mitigates overconfidence by giving higher weights to predictive probabilities, giving less extreme values.

Coagulation and Fragmentation

- For $c \geq 1$, $A_2 \epsilon A_c$ and $A_1 \epsilon A_{|A2|}$ so $|c|$ in A_1 = $|t|$ in A_2.
- To coagulate, merge the tables in A_2 according to the customers in A_1 to make arrangement C. Then split A_2 into sections F_a for each table in C.
- To fragment, fragment each table in C into the smaller tables in F_a.

Coagulation and Fragmentation



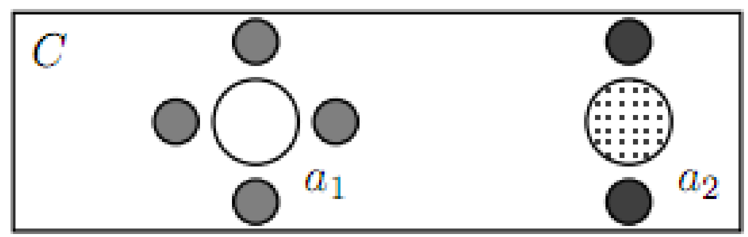


Figure 1: Illustration of the relationship between the restaurants A_1, A_2, C and F_a.

Coagulation and Fragmentation Preserved

Theorem 1 ([4, 5]). *Suppose* $A_2 \in \mathcal{A}_c$, $A_1 \in \mathcal{A}_{|A_2|}$, $C \in \mathcal{A}_c$ *and* $F_a \in \mathcal{A}_{|a|}$ *for each* $a \in C$ *are related as above. Then the following describe equivalent distributions:*
(I) $A_2 \sim \mathrm{CRP}_c(\alpha d_2, d_2)$ *and* $A_1 | A_2 \sim \mathrm{CRP}_{|A_2|}(\alpha, d_1)$.
(II) $C \sim \mathrm{CRP}_c(\alpha d_2, d_1 d_2)$ *and* $F_a | C \sim \mathrm{CRP}_{|a|}(-d_1 d_2, d_2)$ *for each* $a \in C$.

- Proof by math given in paper.
- We can marginalize out all but a linear number of PYPs, giving only a HPYP over prefixes and some ancestors.

Compact Representation

- To keep memory costs down, you typically only store # of customers, # of tables, and size of tables.
  - Can still be too much.
- Instead, only store # of customers and tables, but not table sizes

$$P(\{c_{\mathbf{u}s}, t_{\mathbf{u}s}\}, x_{1:T}) = \left(\prod_{s \in \Sigma} H(s)^{t_{\varepsilon s}}\right) \prod_{\mathbf{u} \in \mathcal{U}} \left(\frac{[\alpha_{\mathbf{u}} + d_{\mathbf{u}}]_{d_{\mathbf{u}}}^{t_{\mathbf{u}\cdot} - 1}}{[\alpha_{\mathbf{u}} + 1]_1^{c_{\mathbf{u}\cdot} - 1}} \prod_{s \in \Sigma} S_{d_{\mathbf{u}}}(c_{\mathbf{u}s}, t_{\mathbf{u}s})\right).$$

Gibbs Sampling

$$P(t_{\mathbf{u}s}|\text{rest}) \propto \frac{[\alpha_{\mathbf{u}} + d_{\mathbf{u}}]_{d_{\mathbf{u}}}^{t_{\mathbf{u}\cdot}-1}}{[\alpha_{\sigma(\mathbf{u})} + 1]_1^{c_{\sigma(\mathbf{u})\cdot}-1}} S_{d_{\mathbf{u}}}(c_{\mathbf{u}s}, t_{\mathbf{u}s}) S_{d_{\sigma(\mathbf{u})}}(c_{\sigma(\mathbf{u})s}, t_{\sigma(\mathbf{u})s}), \tag{11}$$

- t_u, $c_{\sigma(u)}$, and $c_{\sigma(u)s}$ are depedent on t_{us} and c_{us} is determined from c_{us}^{x} and t_{vs} at child restaurants v so this sampler is sufficient.
- Only complication is calculating S.
  - If d is fixed, we can precompute (but takes lots of memory)
  - Can be updated in the sampling, but adds $O(c_{us}^{2})$ per iteration

Re-instantiate Seating Arrangements

- Can alternatively sample a new seating arrangement given t_{us} and c_{us}, then perform Gibbs sampling for new t_{us}
- Doing so will change ancestor restaurants, so they also have to reinstantiate their arrangements
  – Will need to Depth First Search the restaurants, keeping arrangements in memory for all restaurants in the path
  – Has $O(t_{us}c_{us})$, but potentially lower constant

Re-instatiating A

- Re-express A using variables z_i=the # of tables occupied by first i customers, and y_i=label of customer i's table.

$$f(z_i, z_{i-1}) = \begin{cases} i-1-z_i d & \text{if } z_i = z_{i-1}, \\ 1 & \text{if } z_i = z_{i-1}+1, \\ 0 & \text{otherwise.} \end{cases} \quad P(z_{1:c}) = \frac{\prod_{i:z_i=z_{i-1}} (i-1-z_i d)}{S_d(c,t)}.$$

$$P(y_i | z_{1:c}, y_{1:i-1}) = \begin{cases} 1 & \text{if } y_i = i \text{ and } z_i = z_{i-1}+1, \\ \frac{\sum_{j=1}^{i-1} \mathbf{1}(y_j = y_i) - d}{i-1-z_i d} & \text{if } z_i = z_{i-1} \text{ and } y_i \in [i-1]. \end{cases}$$

- Multiplying above, $P(z_{1:c,} y_{1:c})$=P(A), so you can sample $z_{1:C}$ then each y_i sequentially.

Original Gibbs Sampling

- Instead of updating all table info, just find the probability of gaining or losing a table
- Have to compute expensive S, but only for $1 \le t \le t_{us}$ which will be smaller than c_{us}
- Still runs in $O(t_{us}c_{us})$, but now with a large constant for Stirling numbers.

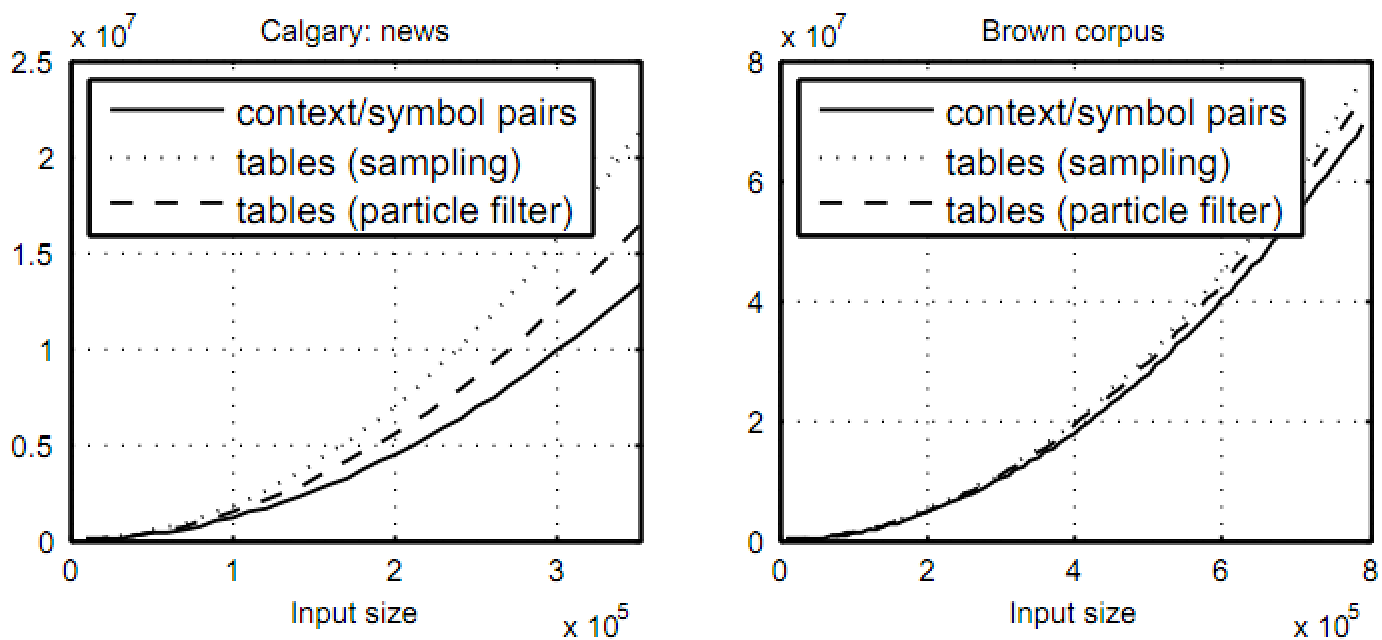
$$P(\text{decrement } t_{\mathbf{u}s}) = \frac{S_{d_\mathbf{u}}(c_{\mathbf{u}s}-1, t_{\mathbf{u}s}-1)}{S_{d_\mathbf{u}}(c_{\mathbf{u}s}, t_{\mathbf{u}s})}. \quad (12)$$

$$P(\text{increment } t_{\mathbf{u}s}) = \frac{(\alpha_\mathbf{u} + d_\mathbf{u} t_{\mathbf{u}\cdot}) P^*_{\sigma(\mathbf{u})}(s)}{(\alpha_\mathbf{u} + d_\mathbf{u} t_{\mathbf{u}\cdot}) P^*_{\sigma(\mathbf{u})}(s) + c_{\mathbf{u}s} - t_{\mathbf{u}s} d_\mathbf{u}}, \quad (13)$$

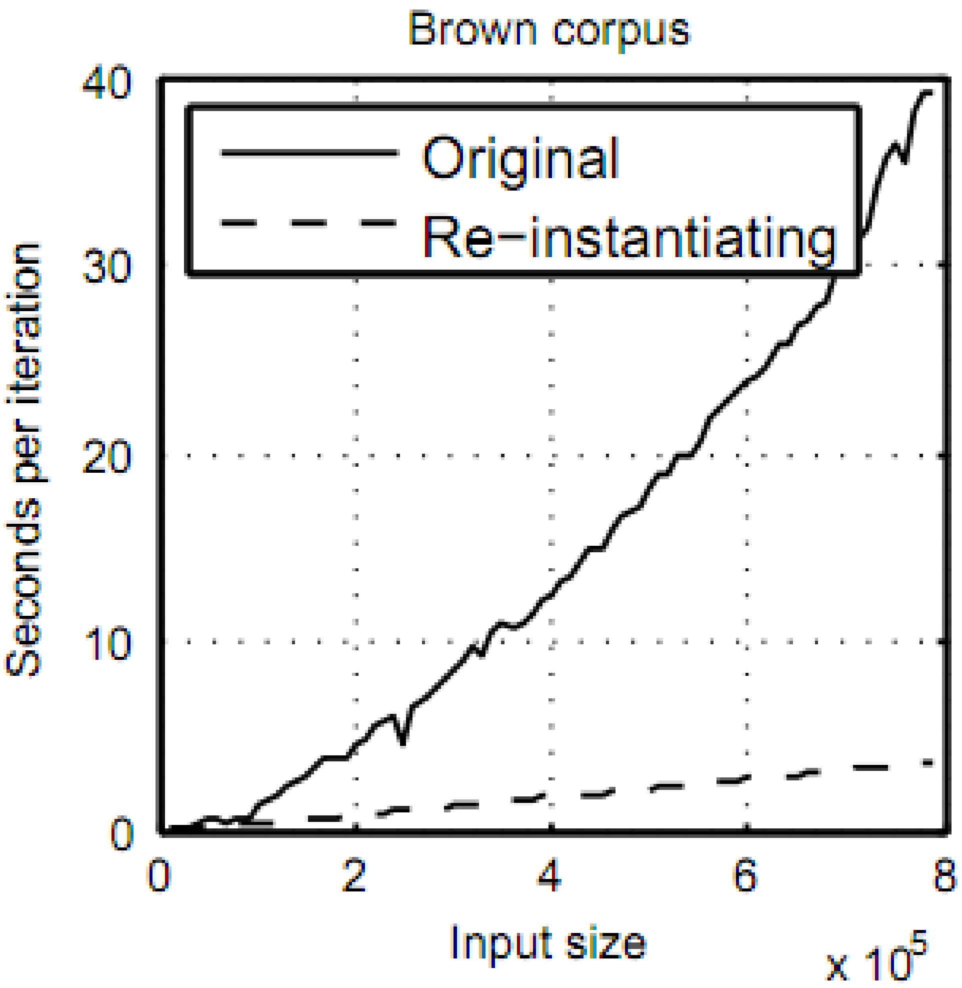
Particle Filtering

- Using the probabilities from before, we can make a Particle Filter
- At each iteration through the sequence $x_{i:T}$, add a new customer according to $s=x_i$, in the context $u=x_{1:i-1}$.

Experiments



Sampling time with Re-instantiation



Concentration Parameter Effects

α	Particle Filter only		Gibbs (1 sample)		Gibbs (50 samples averaged)		Online	
	Fragment	Parent	Fragment	Parent	Fragment	Parent	PF	Gibbs
0	8.45	8.41	8.44	8.41	8.43	8.39	8.04	8.04
1	8.41	8.39	8.40	8.39	8.39	8.38	8.01	8.01
3	8.37	8.37	8.37	8.37	8.35	8.35	7.98	7.98
10	8.33	8.34	8.33	8.33	8.32	8.32	7.95	7.94
20	8.32	8.33	8.32	8.32	8.31	8.31	7.94	7.94
50	8.32	8.33	8.31	8.32	8.31	8.31	7.95	7.95

Questions?