Forgetting Counts

Frank Wood

joint work with Nicholas Bartlett and David Pfau

Columbia University

January 10, 2014
Bayesian Nonparametric Modeling

General Setup

Data

\( \mathcal{X} = \{x_1, x_2, \ldots, x_n \} \)

Model (e.g. density estimation)

\( x_i \sim G \) and \( P(G) \)

Inference

\[
P(X = x | \mathcal{X}) = \int P(X = x | G) dP(G | \mathcal{X})
\]
Bayesian Nonparametric Modeling

Features

- Infinite model “complexity” enables complex inference.
- As $n$ grows, posterior concentrates on empirical distribution.

Considerations

- Inference always function of all data.
- In high dimensions, prior on $G$ plays a key role.

Drawbacks

- Memory and computational complexity grow as a function $n$. 
Example: A Simple PYP Model [12]

Consider

\[ G|d, c, G_0 \sim \mathcal{P}(d, c, G_0) \]
\[ x_i|G \sim G, \quad i = 1, \ldots, n \]

where prediction is performed using

\[ P(x_n|x_1, \ldots, x_{n-1}) = \int P(x_n|G)dP(G|x_1, \ldots, x_{n-1}) \]

Inferring the posterior distribution requires maintaining counts of all observed values.
Example: A Hierarchical PYP Model [15]

Consider a *hierarchical* Pitman-Yor process, such as a smoothed second order Markov model

\[
G[] | d_0, c_0, \mathcal{U} \sim \mathcal{PY}(d_0, c_0, \mathcal{U})
\]
\[
G[x] | d_1, c_1, G[] \sim \mathcal{PY}(d_1, c_1, G[]) \quad \forall x \in \Sigma
\]
\[
G[x,y] | d_2, c_2, G[y] \sim \mathcal{PY}(d_2, c_2, G[y]) \quad \forall x, y \in \Sigma
\]
\[
z_n | G[z_{n-2}, z_{n-1}] \sim G[z_{n-2}, z_{n-1}]
\]

Inference in this model requires maintaining counts of all observed values in all observed contexts of length two. This quantity grows $O(n)$ which becomes impractical for large $n$ (streams, life-long learning, etc.).
Bounded Memory HPYP Inference

- Constant memory requires “forgetting.”
- Sparse Gaussian processes use forgetting strategies to achieve constant time and space inference [9, 5, 13].

Dependent HPYP

- Data generator may change over time or as a function of some other covariate
- Would like single coherent model for such data
- One way of introducing dependence is through forgetting
- Dependent Dirichlet and Pitman-Yor processes have been defined [10, 14, 8, 3, 4]
Outline

Talk Outline

- Review the generative process for the Pitman-Yor process and its relationship with the Ewen’s sampling distribution
- Discuss generative process for the Ewen’s sampling distribution and its properties
- Define a generative process for a dependent HPYP
- Experiments
Pitman-Yor Process

**Model**

\[ G \mid d, c, G_0 \sim \ PY(d, c, G_0) \]
\[ x_i \mid G \sim G, \quad i = 1, \ldots, n \]

**Generative Process**

- Draw a partition of the first \( n \) natural numbers from the two parameter Ewen’s Sampling Distribution (\( ES_n(d, c) \)) [6].
- Assign a value \( \psi_t \sim G_0 \) to part \( t \) of the partition.
- Set \( X_i = \psi_t \) if \( i \) is in the \( t^{th} \) part of the partition.
Ewen’s Sampling Distribution \( (\mathcal{E}S_n(d, c)) \)

**Drawing from \( \mathcal{E}S_n(d, c) \)**

- Assign the integer 1 to part 1 of the partition.
- For \( i = 2, \ldots, n \), assign integer \( i \) to a part drawn from the following distribution:

\[
p(\text{part } t \leq k_{i-1}|\text{previous assignments}) = \frac{n_{t}^{i-1} - d}{i - 1 + c} \\
p(\text{part } k_{i-1} + 1|\text{previous assignments}) = \frac{k^{i-1}d + c}{i - 1 + c}
\]

where \( k^{i-1} \) is the number of partitions to which the first \( i - 1 \) integers have been assigned and \( n_{t}^{i-1} \) is the number of integers \( < i \) assigned to part \( t \).
- Resulting partition follows \( \mathcal{E}S_n(d, c) \) \[11\].
Consistency of $\mathcal{ES}$ Under Deletion

**Property**

If an integer is removed at random from a partition following $\mathcal{ES}_n(d, c)$ the resulting partition of $n - 1$ integers follows $\mathcal{ES}_{n-1}(d, c)$ [11].

**Application**

This consistency property can be exploited in a generative procedure defined using the urn representation to draw samples from a sequence of dependent distributions $\{G^t\}_{t=1}^T$ such that

$$G^t|d, c, G_0 \sim \mathcal{PY}(d, c, G_0) \quad t = 1, \ldots, T$$

$$X^t_i|G^t \sim G^t \quad i = 1, \ldots, N_t, \; t = 1, \ldots, T$$
Dependent PYP

Model

\[ G^t | d, c, G_0 \sim P_Y(d, c, G_0) \]
\[ x_i^t | G^t \sim G, \quad i = 1, \ldots, n^t \]

Generative Process

- For \( t = 1 \), draw a size \( n^1 \) sample as in PYP
- Between time steps, remove objects from the partition
- To generate a sample at time \( t \), add \( n^t \) objects to the partition following to the sequential generative process of the \( ES(d, c) \)
- Assign a random draw from \( G_0 \) to any new parts. Sample elements are given values according to the location of the \( n^t \) newly placed objects
- This process is an extension of GPU DDP [3]
Dependent HPYP

Vocabulary
It is often convenient to think of this generative process as a Chinese restaurant where each part of the partition is a table and each object a customer sitting at a table.

HPYP
- The dependent generative process can be extended to the HPYP
- Deletion may only occur in leaf restaurants
- A leaf restaurant is one such that all restaurants in the subtree stemming downward are empty
Depiction of Dependent HPYP

\[ \psi_1 \rightarrow \psi_2 \rightarrow \psi_3 \]

Parent

Child

\[ \psi_1 \rightarrow \psi_2 \rightarrow \psi_3 \]

Generates \( \mathcal{G}_1 \)

Delete

Generates \( \mathcal{G}_2 \)
## Deletion Points

- The number of customers deleted during the deletion step is independent of the consistency result.
- The number of customers deleted from a given restaurant can be either deterministic or stochastic.
- Only customers in leaf restaurants may be deleted during the deletion step.

## Constant Space

- A generative procedure which deterministically deletes all customers in a chosen leaf restaurant is well specified. Since empty restaurants are not represented during inference, such a model could be used to limit the number of instantiated restaurants and allow for constant space representation.
**Description**

### Compression

- The Sequence Memoizer, an infinite order HPYP, is an effective predictive model for the task of compression [7]
- Files are modeled as a sequence of bytes
- The linear space requirement of the SM prohibits its use on large files

### DHPYP for Compression

- We create a dependent SM for the task of compression
- We control the amount of space required by the model by the number of instantiated restaurants
- Inference is performed using a single particle particle filter, a technique which works well with the SM [7]
Constant Space Sequence Memoizer

- Horizontal axis is memory, measured in restaurants
- Vertical axis is the average negative log loss of the data
- Calgary corpus (14 files, max size = 769kb)
- Two forgetting schemes and two other constant space modeling approaches
- 256 character alphabet
- SM requires 1.16 million instantiated restaurants
Discussion

Future Work

- Other ways to create dependence in a sequence of HPYPs
- More extensive testing with massive data
- Testing of structure discovery with synthetic data
- Properties of single particle sequential Monte Carlo


Bibliography II


