Beam sampling for the infinite HMM

†University of Cambridge, *Gatsby Unit, University College London
Jurgen Van Gael†, Yunus Saatçι†, Yee Whye Teh*, Zoubin Ghahramani†
Context

Sequential data (or time series) are abundant. This talk focuses on discrete time, hidden state models.

**Hidden Markov Model**

- important tool for 4 decades
- applications:
  - Part-Of-Speech Tagging
  - Speech Recognition
  - DNA Sequence Alignment
#### Hidden Markov Model

- **Core**: hidden $K$-state Markov chain
  - initial distribution: $p(s_0 = 1) = 1$
  - transition probability: $p(s_t = j | s_{t-1} = i) = \pi_{ij}$

- **Peripheral**: observation model $y_t \sim F(\phi_{s_t})$
  - e.g. $y_t | s_t \sim \mathcal{N}(\mu_{s_t}, \sigma^2_{s_t})$ or $y_t | s_t \sim \text{Multinomial}(\theta_{s_t})$
  - easy to extend to other observation models

- Parameters of the model are $K, \pi, \phi$
From HMM to Infinite HMM

Classical problem: how to determine the # of states? Can we define a model with an unbounded # of states? If so, can we find a suitable inference algorithm?

→ Yes: the Infinite Hidden Markov Model (or HDP-HMM)
  - [Beal et al. 2002]: introduced the model, approximate sampling.
  - [Teh et al. 2006]: theoretical foundation, introduced Gibbs sampler.
Infinite Hidden Markov Model

\[
\begin{align*}
\beta & \sim \text{Stick}(\gamma), \\
\phi_k & \sim H, \\
\pi_k & \sim \text{Dirichlet}(\alpha \beta), \\
s_t & \sim \text{Multinomial}(\pi_{s_{t-1}}), \quad (s_0 = 1) \\
y_t & \sim F(\phi_{s_t})
\end{align*}
\]

**Parameters**
- observation parameters
- transition matrix

**Hyper parameters**
- controls # states
- prior on observation model parameters
- controls transition matrix row similarity

\[
\begin{align*}
\gamma & \quad \beta \\
\alpha & \quad \pi_k \\
H & \quad \phi_k
\end{align*}
\]

\[
\begin{align*}
s_0 & \quad s_1 & \quad s_2 \\
y_1 & \quad y_2
\end{align*}
\]

\[
k = 1..\infty
\]
Motivation

Inference is the problem of computing posterior distributions.

*Hidden Markov Model*
- Dynamic Programming (= fast)

*Infinite Hidden Markov Model (so far)*
- Gibbs sampling

Recall:
- Gibbs sampling updates one hidden variable at a time.
- Gibbs sampling mixes slow when strong correlations.

→ Trouble: time series often exhibit strong correlations!

*In practice: we need fast inference!*
Can we adapt dynamic programming to our nonparametric model?
Dynamic Programming
Forward-Filtering Backward-Sampling

1. Compute conditional probabilities
   1. Initialize
      \[ p(s_0 = 1) = 1 \]
   2. For each \( t = 1 \ldots T \)
      \[ p(s_t | y_{1:t}) \propto p(y_t | s_t) \sum_{s_{t-1}} p(s_t | s_{t-1}) p(s_{t-1} | y_{1:t-1}) \]

2. Sample hidden states
   1. Sample for time \( T \)
      \[ p(s_T | y_{1:T}) \]
   2. For each \( t = T-1 \ldots 1 \)
      \[ p(s_t | s_{t+1}, y_{1:t}) \propto p(s_{t+1} | s_t) p(s_t | y_{1:t}) \]

\[ O(TK^2) \]
\[ O(TK) \]
Beam Sampling

- Can we use Forward-Filtering Backward-Sampling for the iHMM?
  - No, $O(TK^2)$ with $K \to \infty$ is intractable

- An idea:
  - Truncate transition matrix & use dynamic programming to sample $s$.
  - This is only approximately correct and unnecessary!

Beam Sampling = Adaptive Truncation + Dynamic Programming
Beam Sampling

1. We start with a finite representation of the transition matrix.

2. At every timestep \( t \) we look at the transition taken.

3. We introduce auxiliary variables

\[
u_t \sim \text{Uniform}(0, \pi_{s_{t-1},s_t})
\]

**Key Observation:** since the rows of the transition matrix must sum to 1, only a finite # of sticks > \( u_t \).

4. We only consider paths that use transitions larger than slice.

[Neal, 2003; Walker 2006]
Comment on Auxiliary Variables

The auxiliary variables don’t change the model! We can just marginalize them out and recover the original model.
Beam Sampling Algorithm

1. Initialize hidden states + parameters
2. While (enough samples)
   1. Sample \( p(u | s) \): \( u_t \sim \text{Uniform}(0, \pi_{s_{t-1}, s_t}) \)
   2. Sample \( p(s | u, y) \) using dynamic programming
      1. Initialize DP \( p(s_0 = 1) = 1 \)
      2. For each \( t = 1 \ldots T \)
         \[
p(s_t | y_{1:t}, u_{1:t}) \propto p(y_t | s_t) \sum_{s_{t-1} : u_t < \pi_{s_{t-1}, s_t}} p(s_{t-1} | y_{1:t-1}, u_{1:t-1}).
         \]
   3. Sample \( T \) \( p(s_T | y_{1:T}) \)
   4. Sample \( t = T-1 \ldots 1 \) \( p(s_t | s_{t+1}, y_{1:t}) \propto p(s_{t+1} | s_t)p(s_t | y_{1:t}) \)
3. Resample \( \pi, \phi, \beta, \gamma, \alpha | s \)
Beam Sampling Properties

- The sampler adaptively truncates the infinitely large transition matrix
- The truncation also sparsifies the dynamic program
- Resample the whole sequence $s$
  - Gibbs sampler only changes one hidden state conditioned on all other states
- All parameters need to be instantiated
  - Gibbs sampler can collapse variables
  - Beam sampler can do inference for non-conjugate models
- (Hyper)parameter sampling is identical to Gibbs sampler
Experiment I – HMM Data

*Synthetic data generated by HMM with K=4*

*Strong negative correlation (1-2-3-4-1-2-3-…)*

- **Vague Priors**
  - $\alpha \sim \text{Gamma}(1,1)$
  - $\gamma \sim \text{Gamma}(2,1)$
- **Strong Priors**
  - $\alpha \sim \text{Gamma}(6,15)$
  - $\gamma \sim \text{Gamma}(16,4)$
- **Fixed Priors**
  - $\alpha = 0.4; \gamma = 3.8$

Average # of transitions considered per timestep (i.e. effective complexity of dynamic program) tends to 1.
Experiment II – Changepoint Detection

Well Log (NMR Response) – Change point Detection
- 4050 noisy NMR response measurements
- Output model is Student-t with known scale

Beam sampler output of iHMM after 8000 iterations:
Experiment II – Changepoint Detection

What is the probability of two data points in same cluster?

- Left: average over first 5 samples
- Right: average over last 30 samples datapoints

Note: 1) gray areas for beam; 2) slower mixing for Gibbs
Experiment III – Text Prediction

Alice in Wonderland

- training data: 1000 characters from 1st chapter
- 35 possible output characters
- testing data: 1000 subsequent characters

VB-HMM:
- Transition matrix: Dirichlet(4/K, ..., 4/K)
- Emission matrix: Dirichlet(0.3)

iHMM:
- $\alpha \sim \text{Gamma}(4,1)$
- $\gamma \sim \text{Gamma}(1,1)$
- $H \sim \text{Dirichlet}(0.3)$
Conclusion

- Inference based on dynamic programming
  - Adaptive truncation = sampling from true posterior
  - Adaptive truncation = dynamic programming speedup
- Inference for non-conjugate models
- iHMM can be good alternative for (VB)-HMM

The Beam sampler makes inference in the iHMM fast enough for practical applications.
Thank You!

Questions?
Hidden Markov Model

- **Likelihood**

\[
p(y_1, \cdots, y_T, s_1, \cdots, s_T | \bm{\pi}, \phi) = \prod_{i=1}^{T} p(s_{t} | s_{t-1}) p(y_{t} | s_{t})
\]

\[
= \prod_{i=1}^{T} \pi_{s_{t-1}, s_{t}} F(\phi_{s_{t}})
\]

- **Example**
HMM versus iHMM

HMM is fully specified given
• K parameters
• K by K transition matrix

<table>
<thead>
<tr>
<th>$\phi$</th>
<th>$\phi_1$</th>
<th>$\phi_2$</th>
<th>$\phi_3$</th>
<th>$\cdots$</th>
<th>$\phi_K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi$</td>
<td>$\pi_{11}$</td>
<td>$\pi_{12}$</td>
<td>$\cdots$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\pi_{12}$</td>
<td>$\cdots$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\pi_{KK}$</td>
</tr>
</tbody>
</table>
HMM versus iHMM

iHMM is fully specified given an infinite number of DP’s ?!?.

<table>
<thead>
<tr>
<th>$\phi$</th>
<th>$\phi_1$</th>
<th>$\phi_2$</th>
<th>$\phi_3$</th>
<th>$\cdots$</th>
<th>$\cdots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\pi$</td>
<td>$\pi_{11}$</td>
<td>$\pi_{12}$</td>
<td>$\cdots$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\pi_{12}$</td>
<td>$\cdots$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\cdot$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>