

Nonparametric Bayesian times series models: infinite HMMs and beyond

Zoubin Ghahramani
University of Cambridge

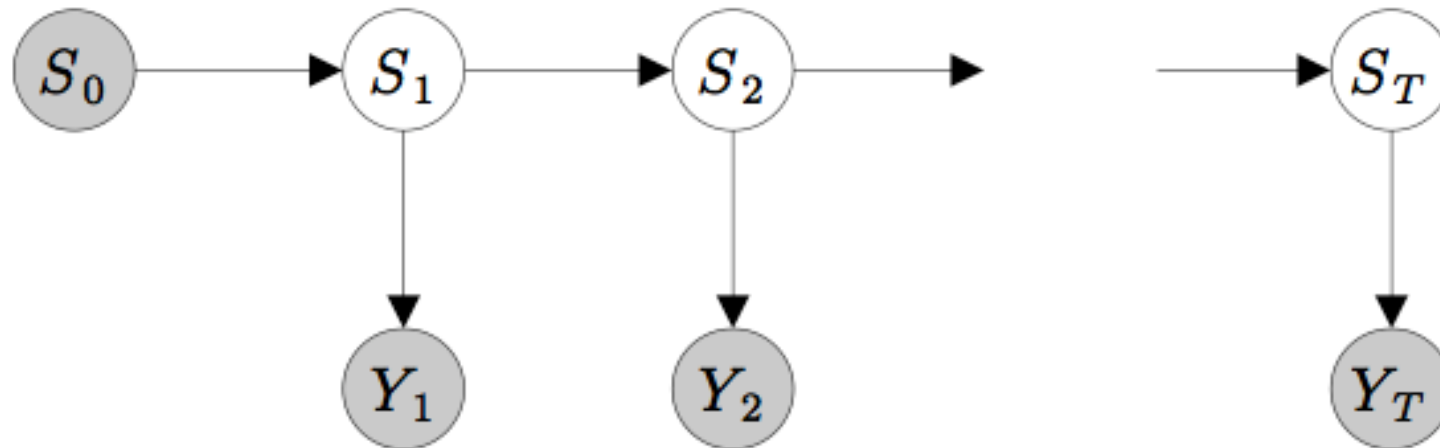
joint work with

Jurgen van Gael, Yee Whye Teh, Yunus Saatci

Motivation

- Broadly two classes of time series models:
 - fully observed models (e.g. AR, n-gram)
 - hidden state models (e.g. HMM, state-space models)
- Hidden Markov models (HMMs) are widely used, but how do we choose the number of hidden states?
- A non-parametric Bayesian approach: infinite HMMs.
- A single discrete state variable is a poor representation of the history. Can we do better?
- Factorial HMMs
- Can we make Factorial HMMs non-parametric?
- infinite factorial HMMs and the Markov Indian Buffet Process

Hidden Markov Models



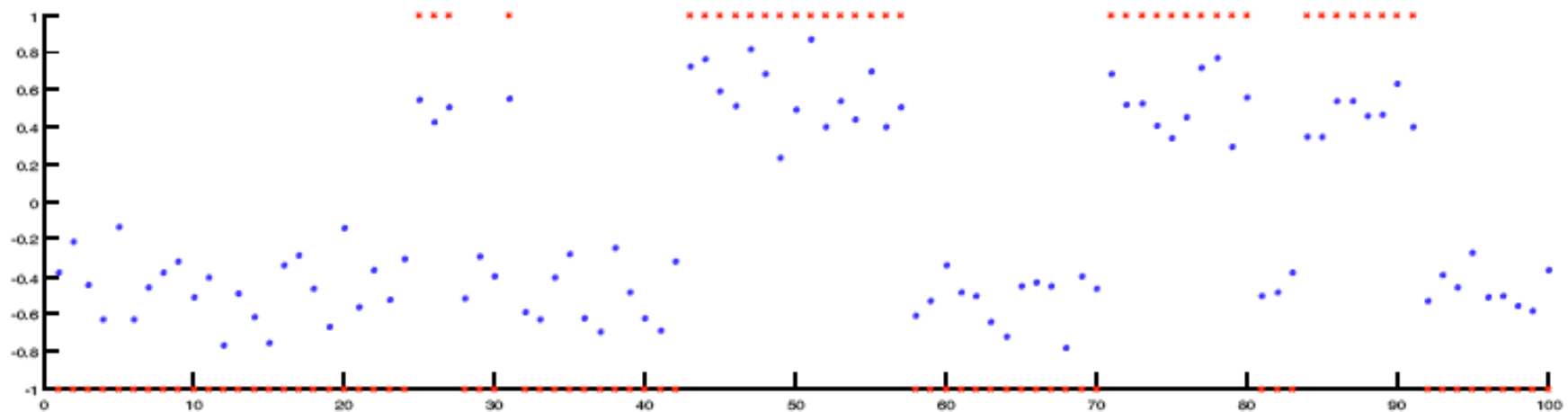
- Core: hidden K -state Markov chain
 - initial distribution $p(s_0 = 1) = 1$
 - transition probability $p(s_t = j | s_{t-1} = i) = \pi_{ij}$
- Peripheral: observation model $y_t \sim F(\phi_{s_t})$
- Parameters of the model are K, π, ϕ

Hidden Markov Models

- Likelihood

$$p(y_1, \dots, y_T, s_1, \dots, s_T | \pi, \phi) = \prod_{i=1}^T p(s_t | s_{t-1}) p(y_t | s_t)$$
$$= \prod_{i=1}^T \pi_{s_{t-1}, s_t} F(\phi_{s_t})$$

- Example



Choosing the number of hidden states

- How do we choose K , the number of hidden states, in an HMM?
- Can we define a model with an unbounded number of hidden states, and a suitable inference algorithm?

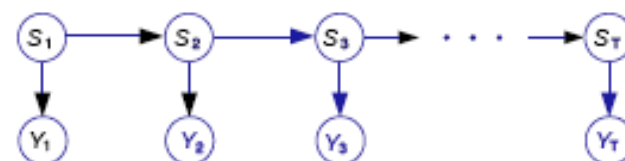
Part I

The Infinite Hidden Markov Model

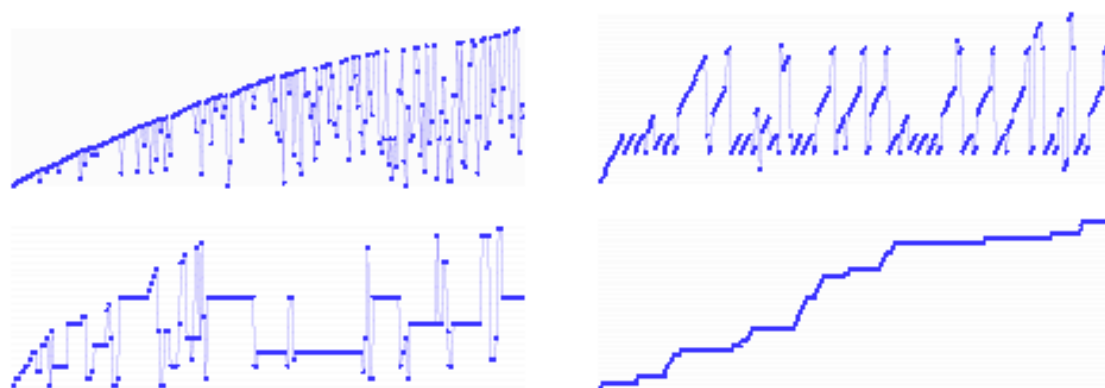
Infinite Hidden Markov models

Hidden Markov models (HMMs) can be thought of as time-dependent mixtures.

In an HMM with K states, the transition matrix has $K \times K$ elements.

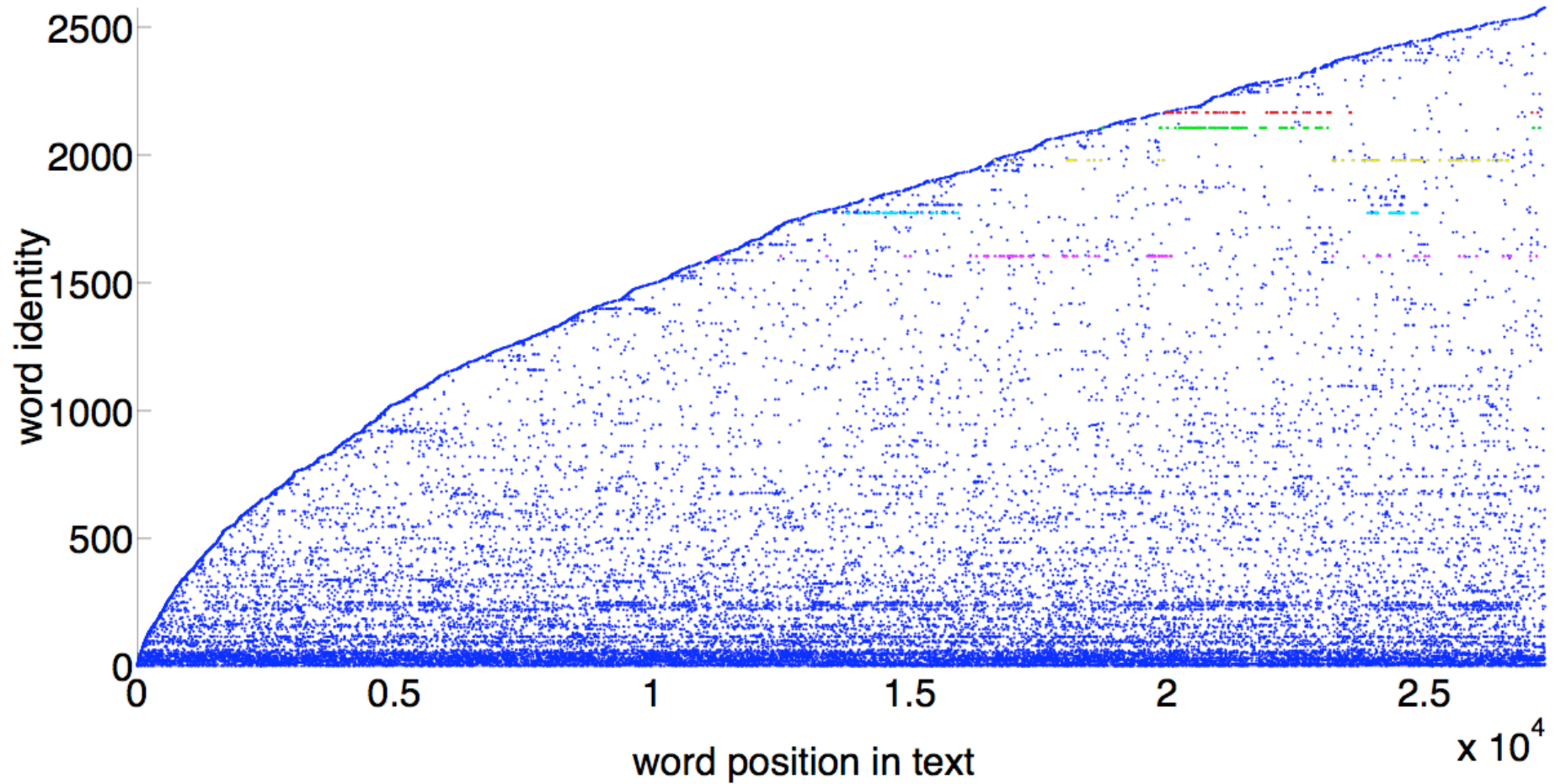


We let $K \rightarrow \infty$, this results in an iHMM.

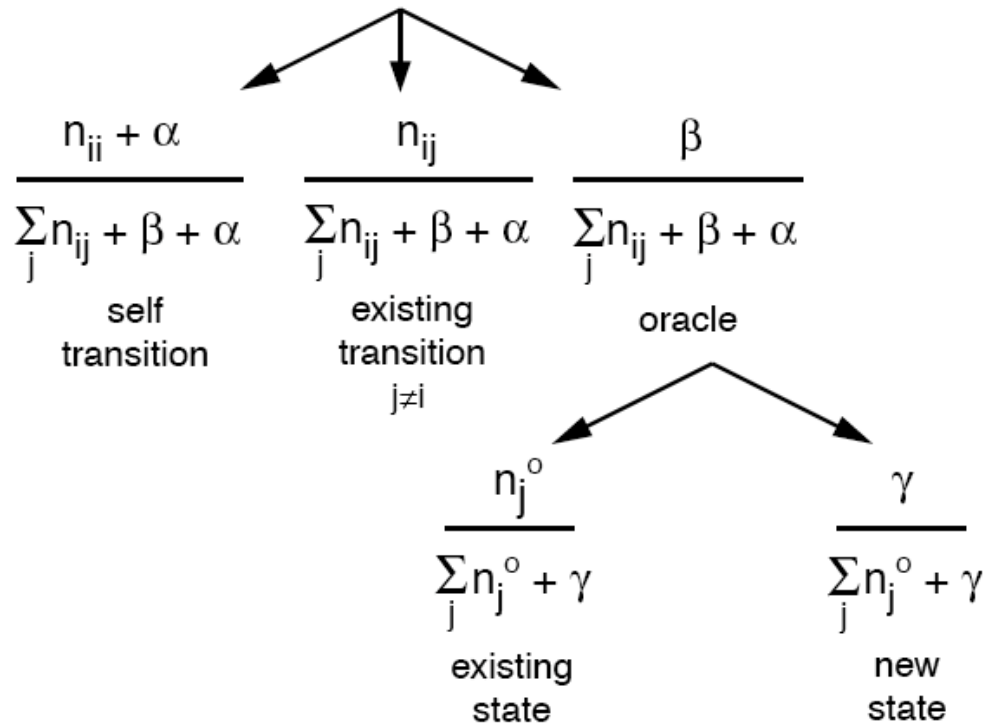


- Introduced in (Beal, Ghahramani and Rasmussen, 2002).
- Teh, Jordan, Beal and Blei (2005) showed that iHMMs can be derived from hierarchical Dirichlet processes, and provided a more efficient Gibbs sampler.
- We have recently derived a much more efficient sampler based on Dynamic Programming (Van Gael, Saatci, Teh, and Ghahramani, 2008).

Alice in Wonderland



Hierarchical Urn Scheme for generating transitions in the iHMM (2002)

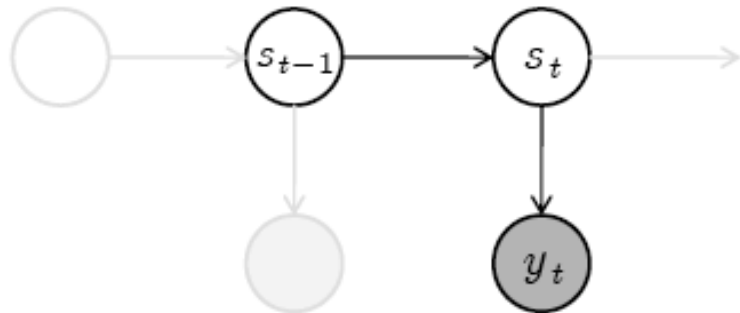


- n_{ij} is the number of previous transitions from i to j
- α , β , and γ are hyperparameters
- prob. of transition from i to j proportional to n_{ij}
- with prob. proportional to $\beta\gamma$ jump to a **new state**

Relating iHMMs to DPMs

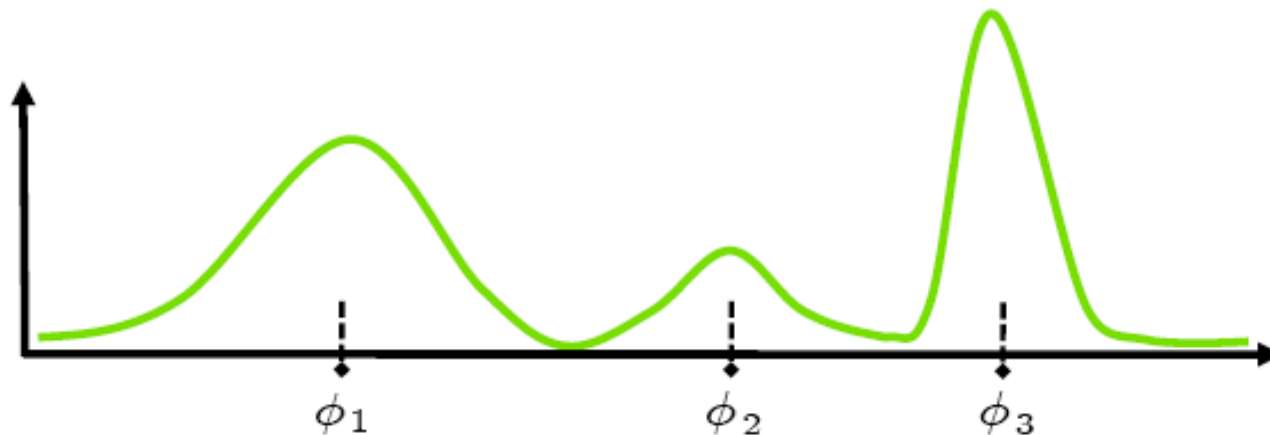
- The infinite Hidden Markov Model is closely related to Dirichlet Process Mixture (DPM) models
- This makes sense:
 - HMMs are time series generalisations of mixture models.
 - DPMs are a way of defining mixture models with countably infinitely many components.
 - iHMMs are HMMs with countably infinitely many states.

HMMs as sequential mixtures



$$\begin{aligned} p(y_t | s_{t-1} = k) &= \sum_{s_t=1}^K p(s_t | s_{t-1} = k) p(y_t | s_t) \\ &= \sum_{s_t=1}^K \pi_{k, s_t} F(\phi_{s_t}) \end{aligned}$$

What is conditional distribution of y_t ?



$p(y_t | s_{t-1} = k)$ is a mixture distribution with K components.

Infinite Hidden Markov Models

- We want HMM in the limit of $K \rightarrow \infty$

Dirichlet Process

- Specifies a distribution over distributions
- We write $G_k \sim \text{DP}(\alpha, H)$ with
 - concentration parameter α
 - base distribution H
- A DP is discrete with probability 1

$$G_k(\phi) = \sum_{k'=1}^{\infty} \pi_{k'} \delta_{\phi_{k'}}(\phi) \quad \forall k' : \phi_{k'} \sim H,$$

- A DP specifies both mixture weights and parameters

Infinite Hidden Markov Models

- Idea: introduce DP's
 - identify mixture weights with HMM transitions
 - identify base distribution draws with observation model parameters

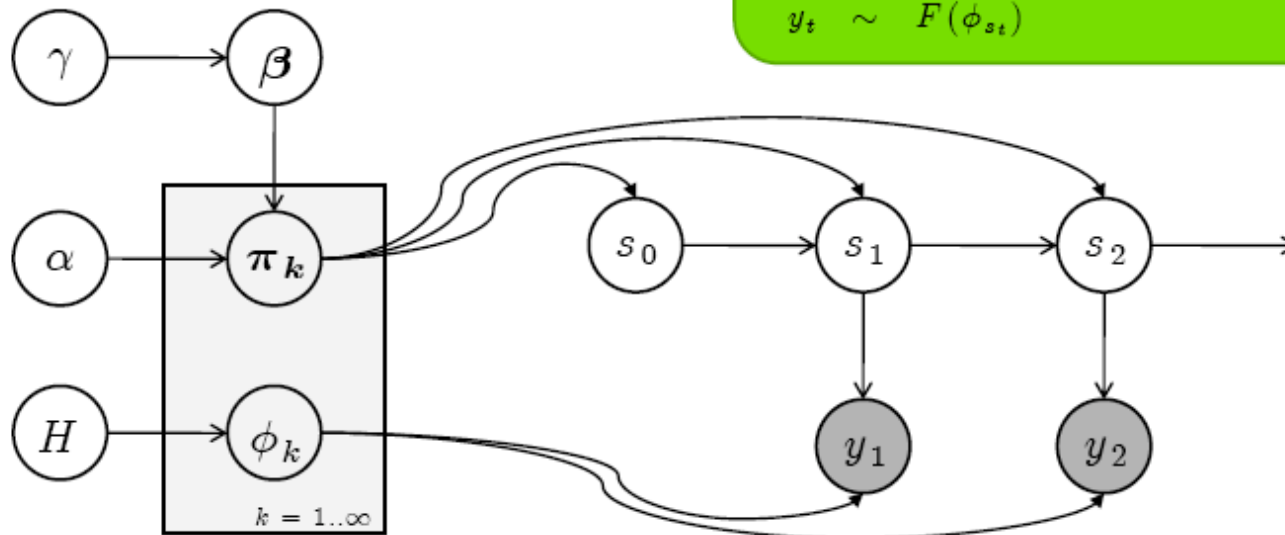
$$p(y_t | s_{t-1} = k) = \sum_{s_t=1}^K \pi_{k,s_t} F(\phi_{s_t})$$
$$G_k(\phi) = \sum_{k'=1}^{\infty} \pi_{k,k'} \delta_{\phi_{k,k'}}(\phi)$$

Infinite Hidden Markov Models

- Recall $G_0(\phi) = \sum_{k'=1}^{\infty} \beta_{k'} \delta_{\phi_{k'}}(\phi)$ $\forall k' : \phi_{k'} \sim H$, $G_k(\phi) = \sum_{k'=1}^{\infty} \pi_{k,k'} \delta_{\phi_{k'}}(\phi)$

- Generative Model for iHMM

$\beta \sim \text{Stick}(\gamma),$
 $\phi_k \sim H,$
 $\pi_k \sim \text{Dirichlet}(\alpha\beta),$
 $s_t \sim \text{Multinomial}(\pi_{s_{t-1}}), \quad (s_0 = 1)$
 $y_t \sim F(\phi_{s_t})$



Teh, Jordan, Beal and Blei (2005) derived iHMMs in terms of Hierarchical Dirichlet Processes.

Efficient inference in iHMMs?

Inference and Learning in HMMs and iHMMs

- **HMM** inference of hidden states $p(s_t | y_1 \dots y_T, \theta)$:
 - forward backward = dynamic programming = belief propagation
- **HMM** parameter learning:
 - Baum Welch = expectation maximization (EM), or Gibbs sampling (Bayesian)
- **iHMM** inference and learning, $p(s_t, \theta | y_1 \dots y_T)$:
 - Gibbs Sampling
- This is unfortunate: Gibbs can be very slow for time series!
- Can we use dynamic programming?

Dynamic Programming in HMMs

Forward Backtrack Sampling

1. Compute conditional probabilities

1. Initialize

$$p(s_0 = 1) = 1$$

$$O(TK^2)$$

2. For each $t = 1 \dots T$

$$p(s_t | y_{1:t}) \propto p(y_t | s_t) \sum_{s_{t-1}} p(s_t | s_{t-1}) p(s_{t-1} | y_{1:t-1})$$

2. Sample hidden states

1. Sample for time T

$$p(s_T | y_{1:T})$$

$$O(TK)$$

2. For each $t = T-1 \dots 1$

$$p(s_t | s_{t+1}, y_{1:t}) \propto p(s_{t+1} | s_t) p(s_t | y_{1:t})$$

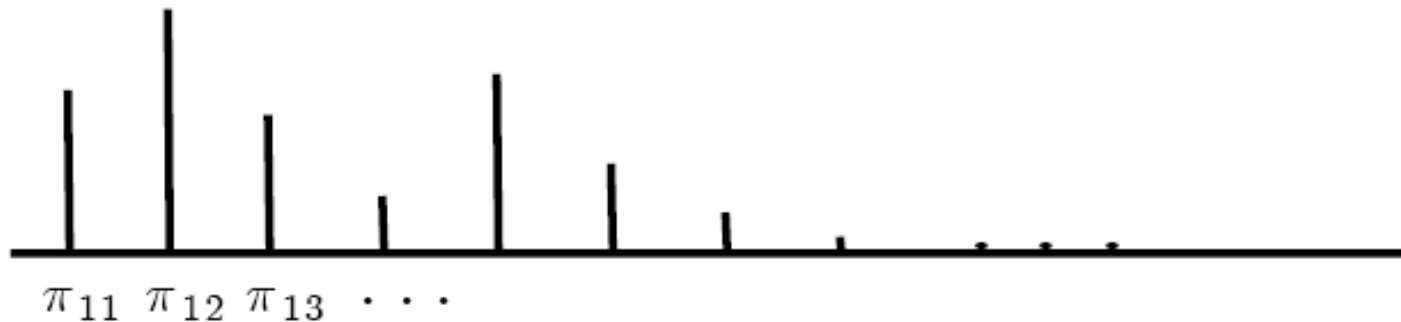
Beam Sampling

- Can we use Forward-Backtrack for iHMM?
 - ➔ No, $O(TK^2)$ with $K \rightarrow \text{infinity}$ is intractable
- A (bad?) idea:
 - Truncate transition matrix
 - Use dynamic programming to sample \mathbf{s}
- This is only approximately correct.

➔ Beam Sampling = Slice Sampling
+
Dynamic Programming

Beam Sampling

- Each G_k can be represented as



- Let us introduce an auxiliary variable

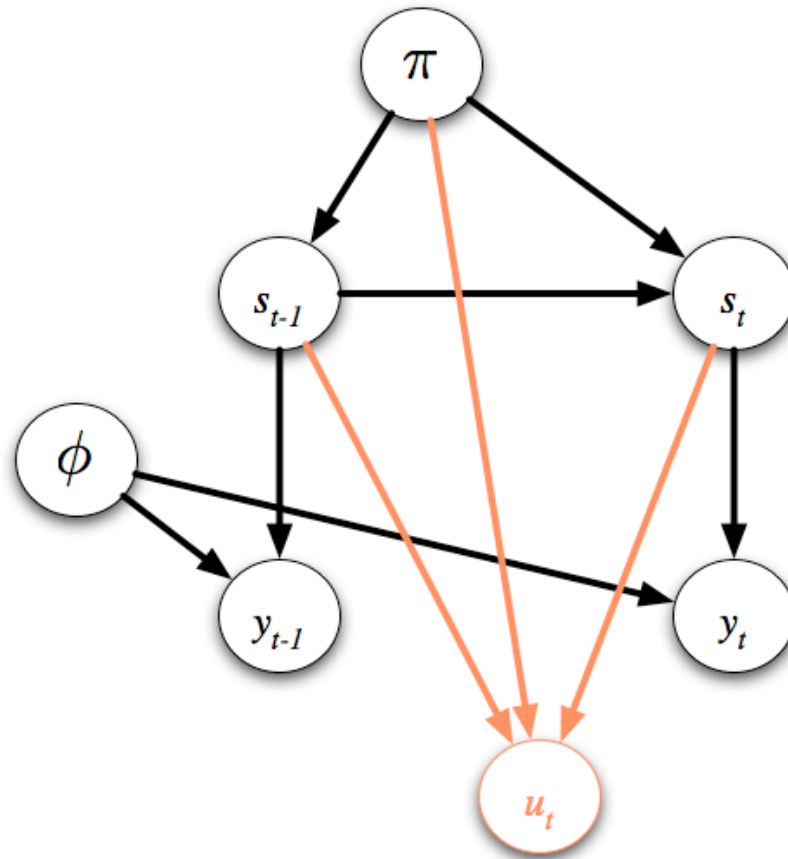
$$u_t \sim \text{Uniform}(0, \pi_{s_{t-1}, s_t})$$

- u_t partitions up $G_{s_{t-1}}$



Key Observation: since π must sum to 1, only a finite # of sticks $> u_t$.

Auxiliary variables



Note: adding u variables, does not change distribution over other vars.

Beam Sampling

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 $t = 1 \dots T$
$$p(s_t | y_{1:t}, u_{1:t}) \propto p(y_t | s_t) \sum_{s_{t-1}: u_t \leq \pi_{s_{t-1}, s_t}} p(s_{t-1} | y_{1:t-1}, u_{1:t-1})$$
 3. Sample T $p(s_T | y_{1:T})$
 4. Sample $t = T-1 \dots 1$ $p(s_t | s_{t+1}, y_{1:t}) \propto p(s_{t+1} | s_t) p(s_t | y_{1:t})$
3. Resample $\pi, \phi, \mathbf{beta}, \gamma, \alpha | s$

Beam Sampling Properties

- The slice sampler adaptively truncates the infinitely large transition matrix
- Dynamic program allows us to resample the whole sequence \mathbf{s}
 - Gibbs sampler only changes one hidden state conditioned on all other states
- The dynamic program needs all parameters to be instantiated
 - Gibbs sampler can collapse variables
 - Beam sampler can do inference for non-conjugate models
- (Hyper)parameter sampling is identical to Gibbs sampling

Experiment I - HMM data

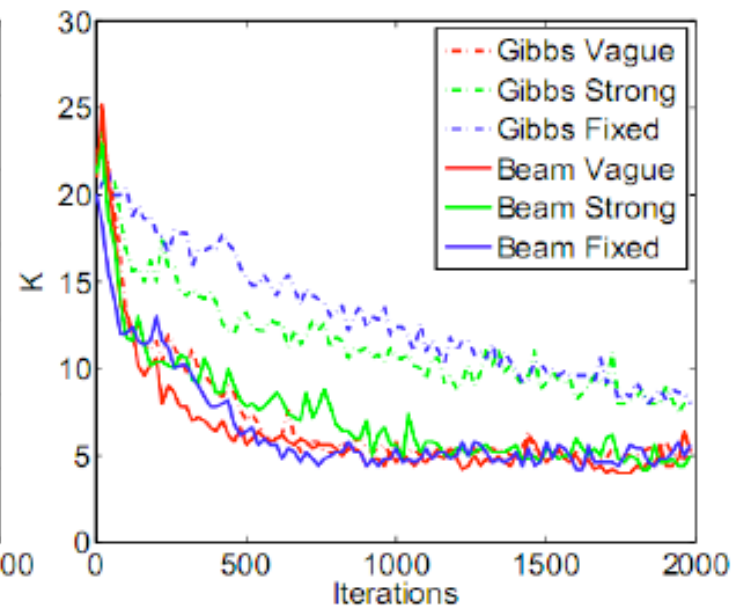
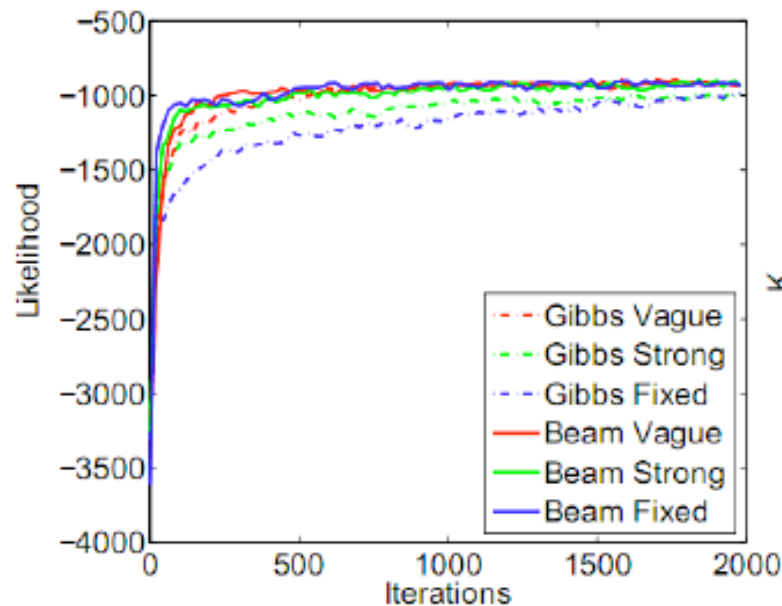
Synthetic data generated by HMM with $K=4$

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

Transition Matrix



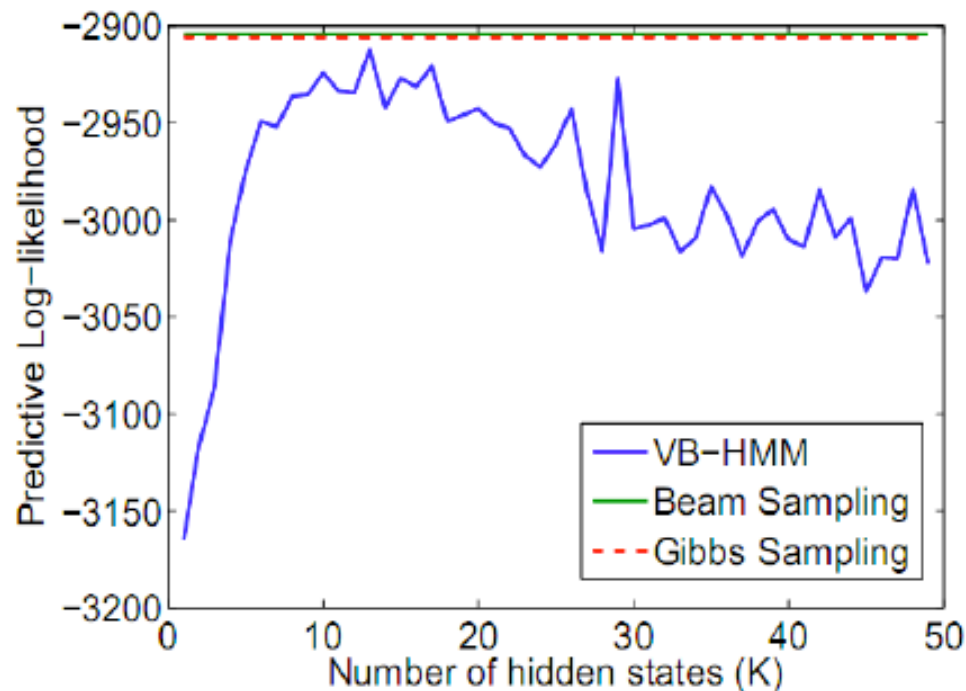
Emission Matrix



Experiment II - 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($4/K, \dots, 4/K$)
- Emission matrix: Dirichlet(0.3)

iHMM:

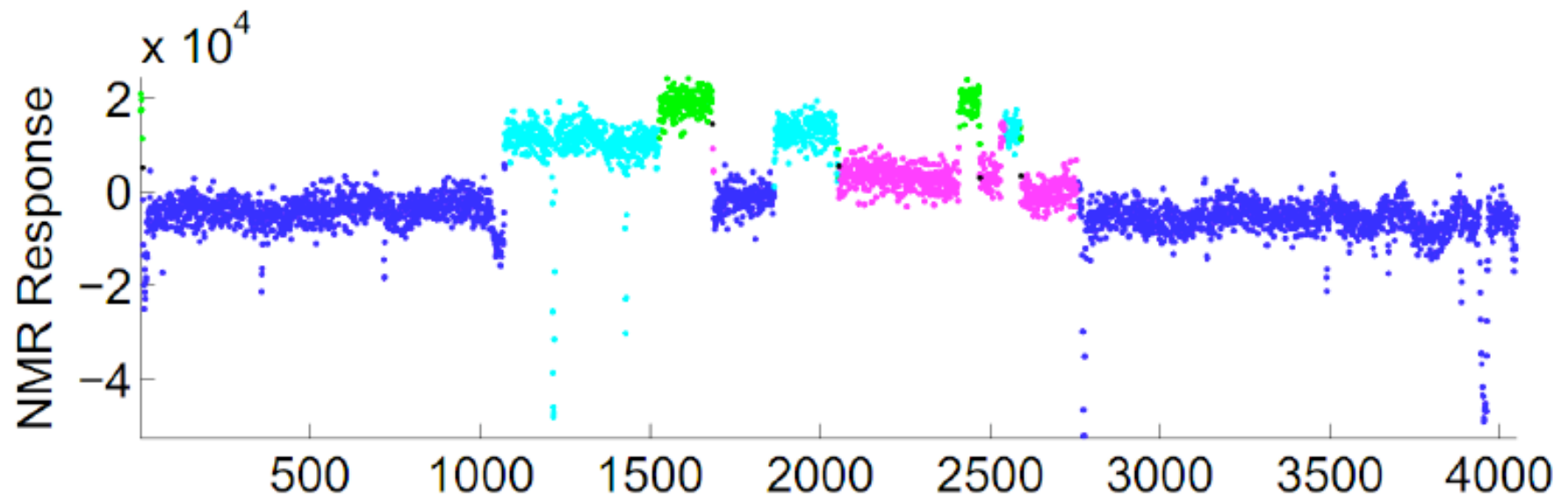
- $\alpha \sim \text{Gamma}(4, 1)$
- $\gamma \sim \text{Gamma}(1, 1)$
- $H \sim \text{Dirichlet}(0.3)$

Experiment III - Change point 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:

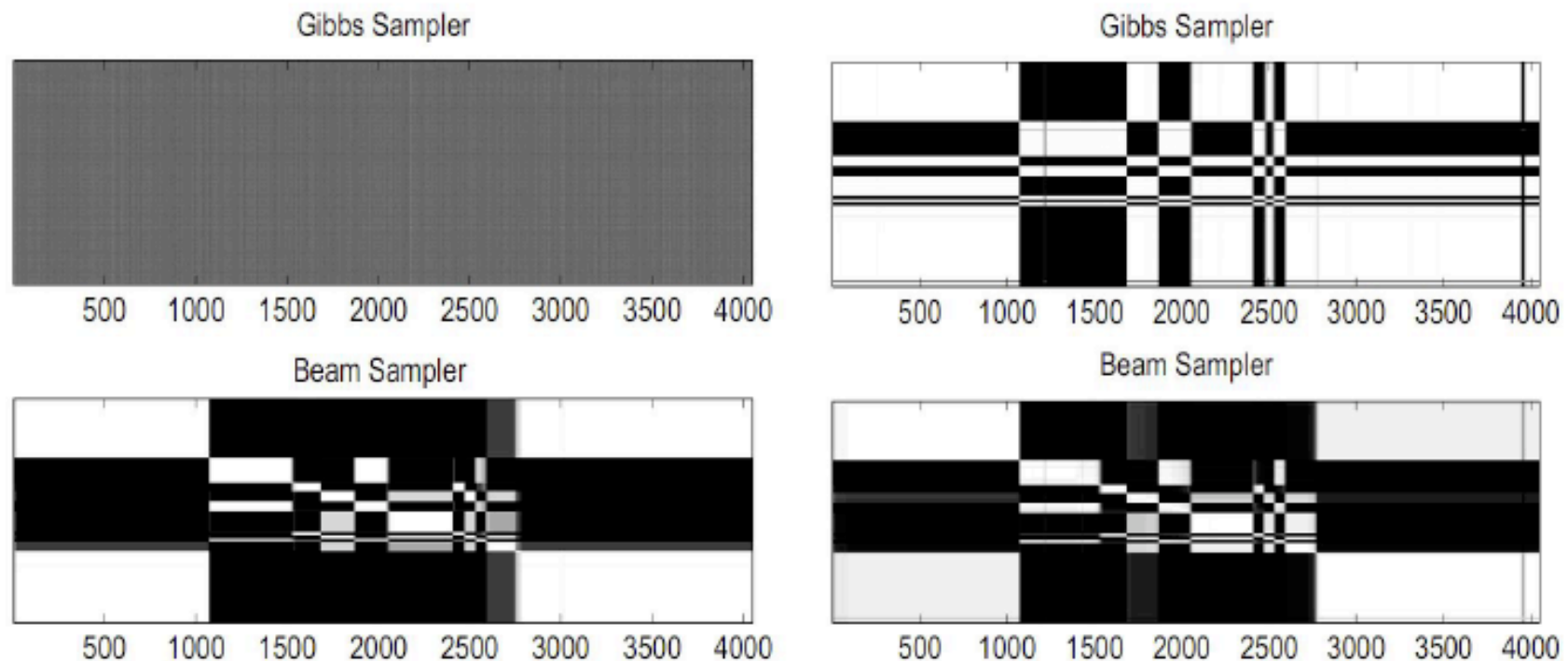


Experiment III - Changepoint Detection

What is probability of two data points in same cluster?

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

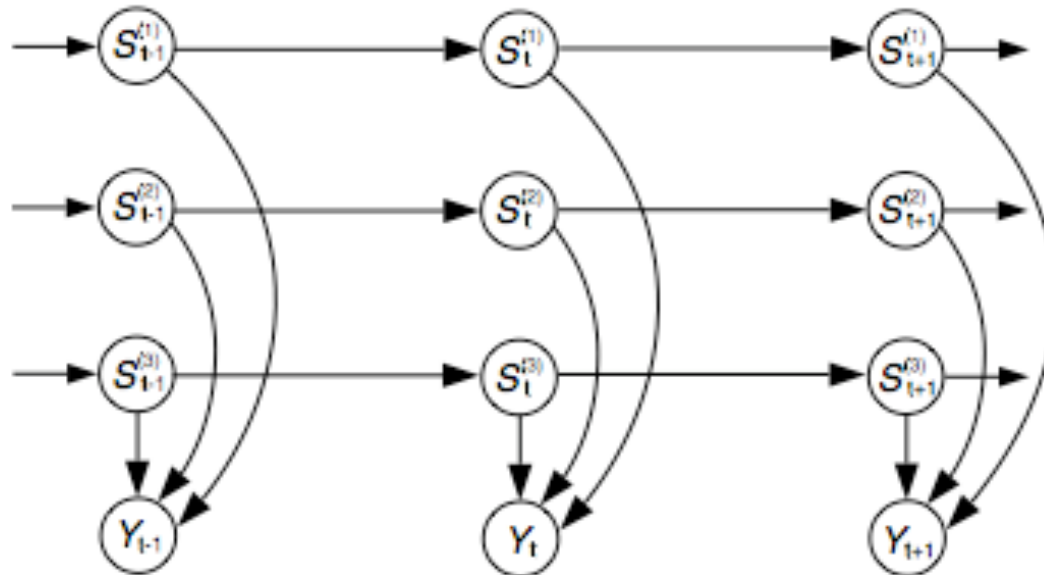
Note: 1) gray areas for beam; 2) slower mixing for Gibbs



Part II

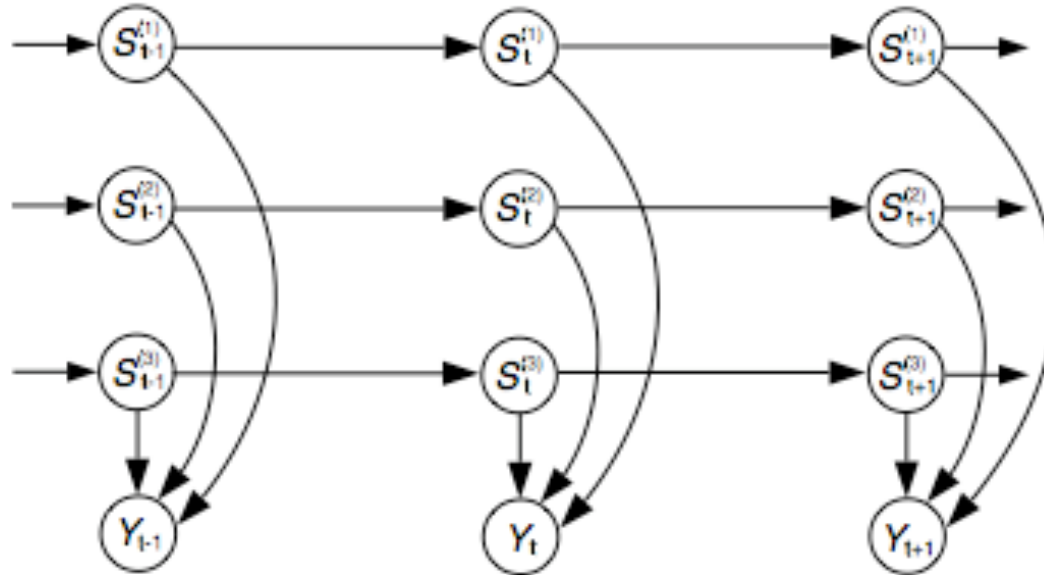
- Hidden Markov models represent the entire history of a sequence using a single state variable s_t
- This seems restrictive...
- It seems more natural to allow many hidden state variables, a “distributed representation” of state.
- ...the *Factorial Hidden Markov Model*

Factorial HMMs



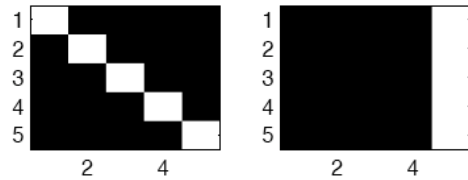
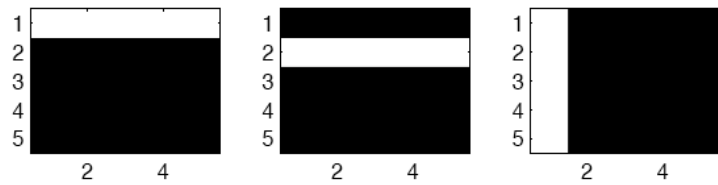
- Factorial HMMs (Ghahramani and Jordan, 1997)
- A kind of dynamic Bayesian network.
- Inference using variational methods or sampling.
- Have been in a variety of applications (e.g. condition monitoring, biological sequences, speech recognition).

From factorial HMMs to infinite factorial HMMs?

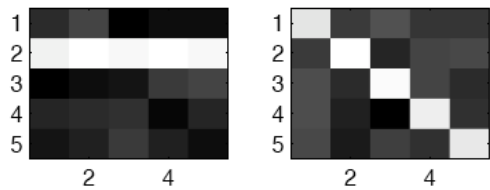
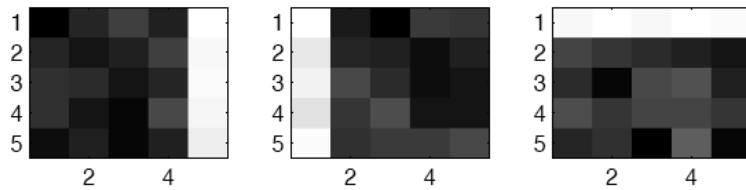


- A non-parametric version where the number of chains is unbounded?
- In infinite factorial HMM (ifHMM) each chain is binary (van Gael, Teh, and Ghahramani, *submitted*).
- Based on the Markov extension of the Indian Buffet Process (IBP).

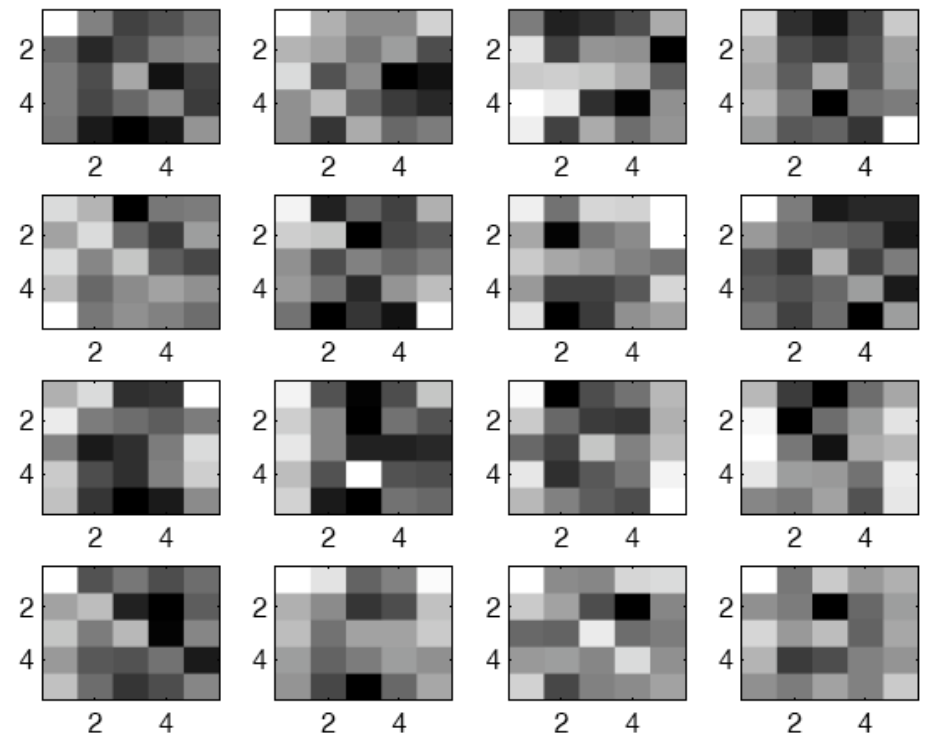
ifHMM Preliminary Experiment: Bars-in-time



Ground
truth

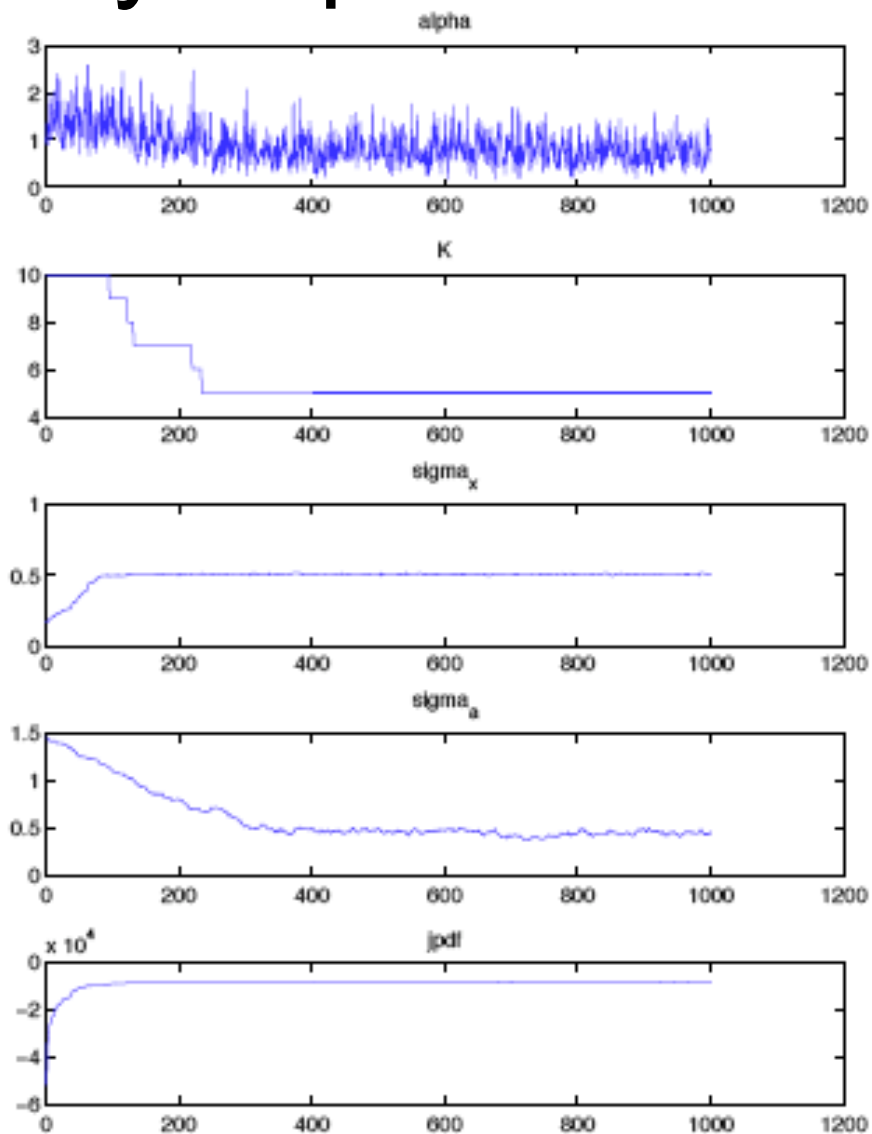


Inferred



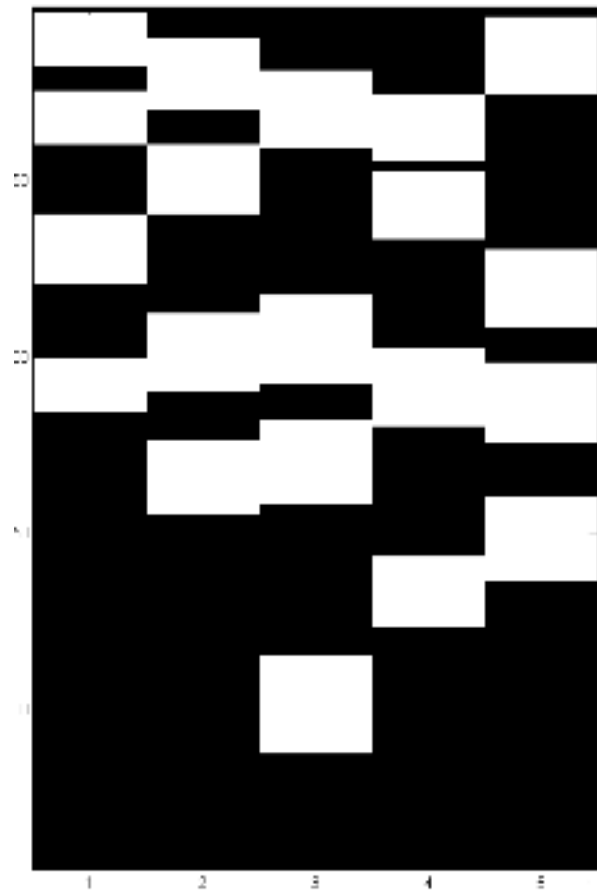
Data

ifHMM Preliminary Experiment: Bars-in-time

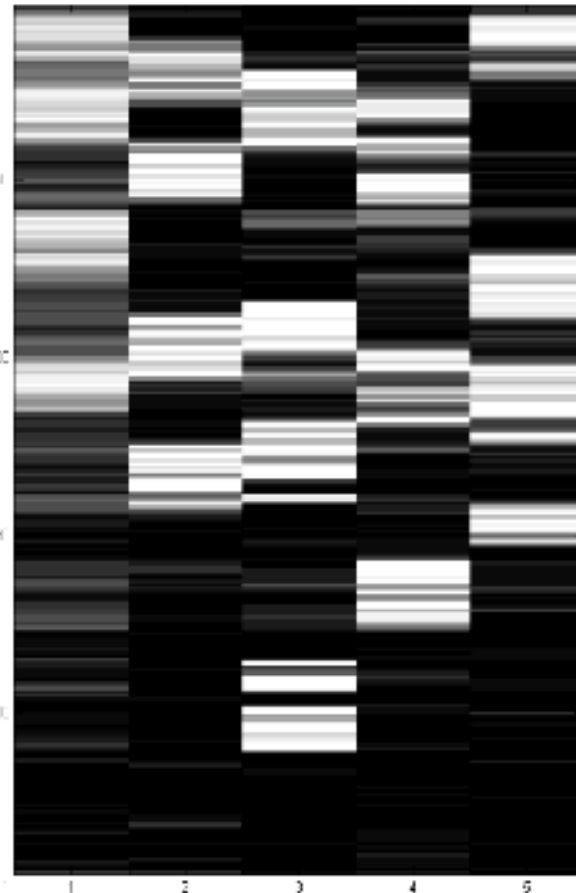


ICA iFHMM

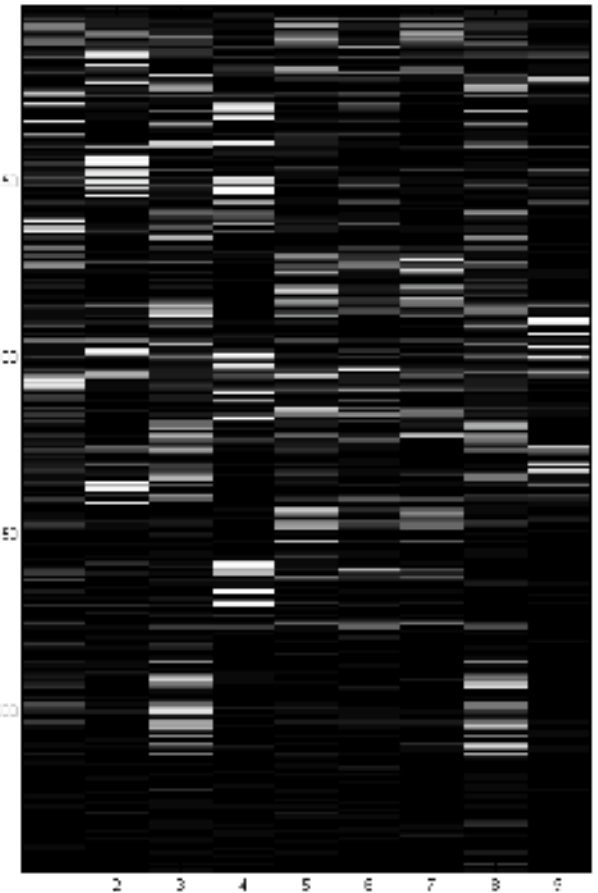
(more signals than sources)



True



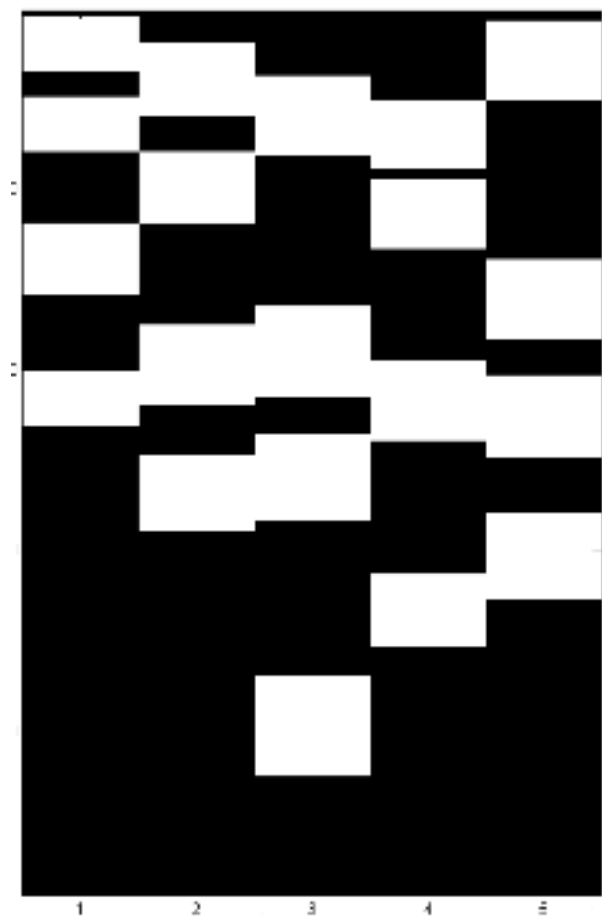
ICA iFHMM



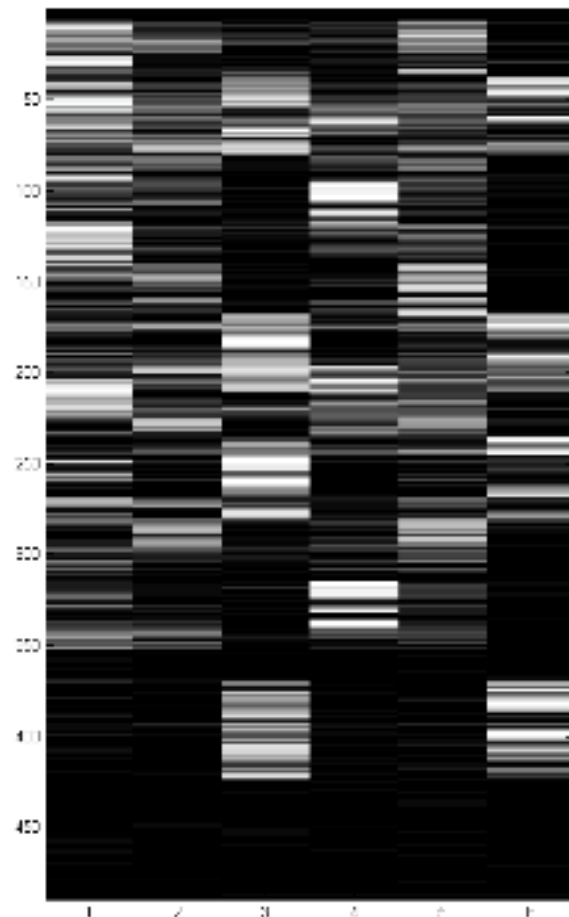
iICA

ICA iFHMM

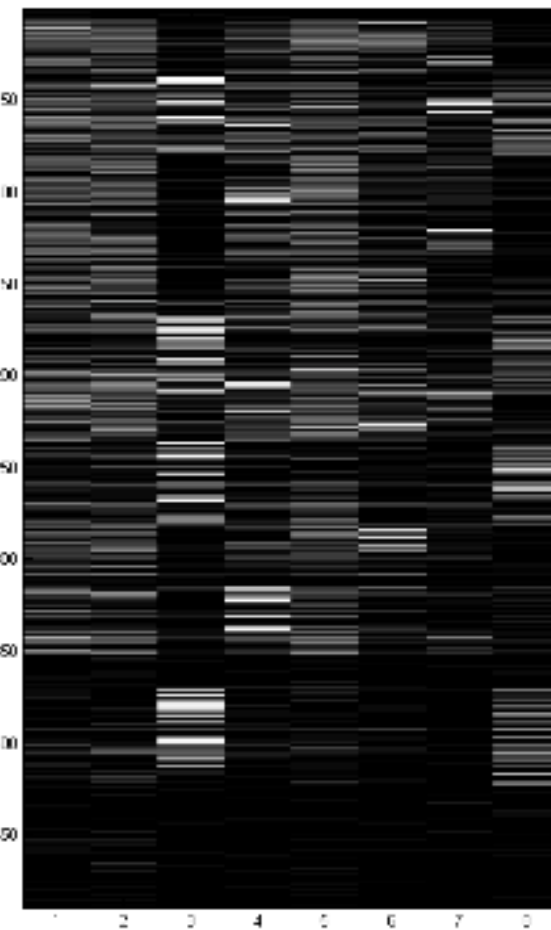
(fewer signals than sources)



True

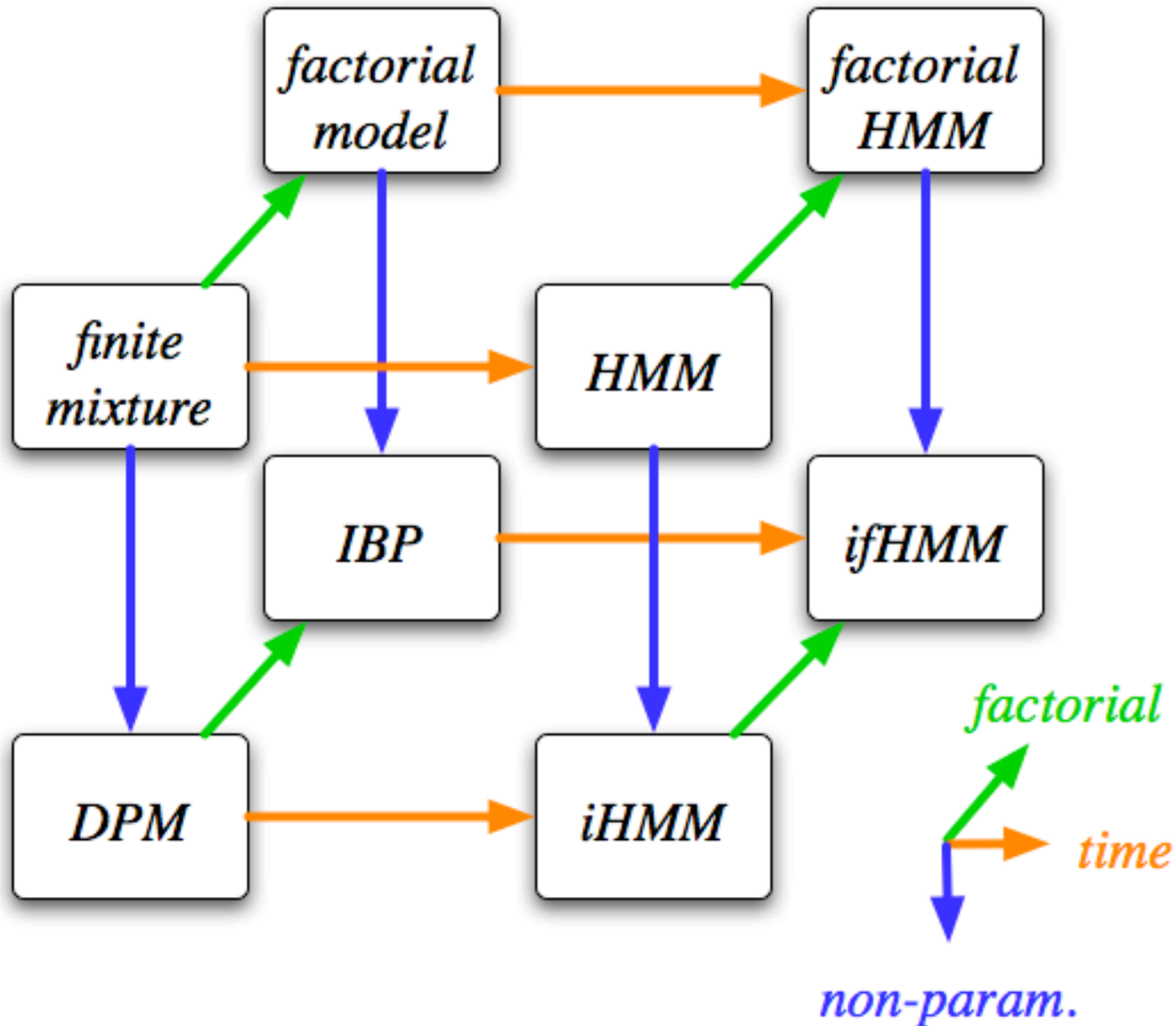


ICA iFHMM



iICA

The Big Picture



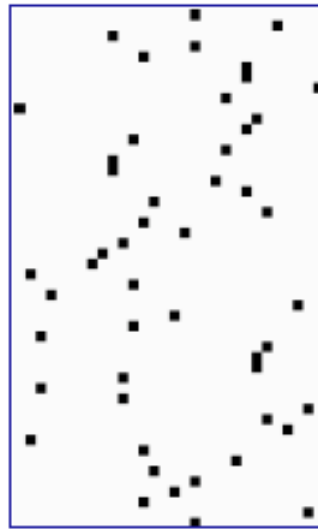
Conclusion

- HMMs have been widely used.
- **iHMMs** provide a non-parametric version where the number of states is not bounded a priori.
- **Beam sampling** provides an efficient exact dynamic programming-based MCMC method.
- **ifHMMs** extend iHMMs to multiple state variables in parallel.
- **Future directions:** new models, fast algorithms, and compelling applications.

Appendix:

Indian Buffet Process (IBP)

Binary Matrix Representation of Clustering



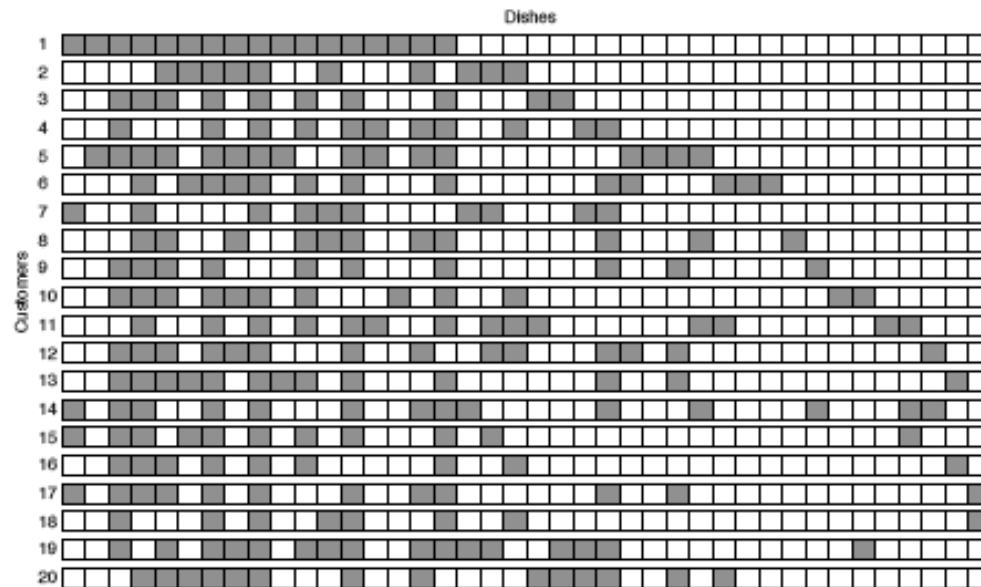
- Rows are data points
- Columns are clusters
- Since each data point is assigned to one and only one cluster...
- ...the rows sum to one.

Binary Latent Feature Matrices



- Rows are data points
- Columns are latent **features**
- We can think of **infinite** binary matrices...
...where each data point can now have *multiple* features, so...
...the rows can sum to more than one.

Indian Buffet Process



“Many Indian restaurants in London offer lunchtime buffets with an apparently infinite number of dishes”



- First customer starts at the left of the buffet, and takes a serving from each dish, stopping after a $\text{Poisson}(\alpha)$ number of dishes as her plate becomes overburdened.
- The n th customer moves along the buffet, sampling dishes in proportion to their popularity, serving himself with probability m_k/n , and trying a $\text{Poisson}(\alpha/n)$ number of new dishes.
- The customer-dish matrix is our feature matrix, \mathbf{Z} .

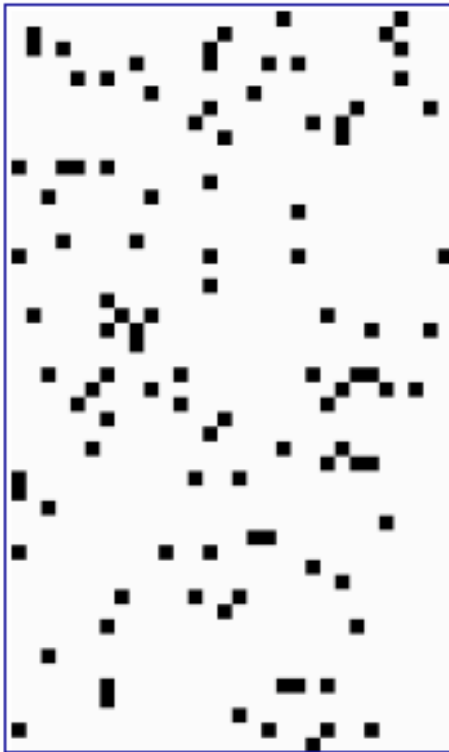
(Griffiths and Ghahramani, 2005)

Indian Buffet Process

$z_{nk} = 1$ means object n has feature k :

$$z_{nk} \sim \text{Bernoulli}(\theta_k)$$

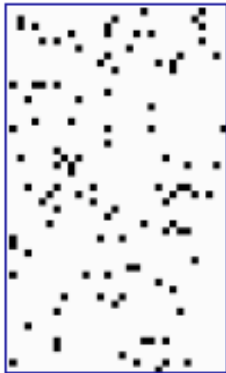
$$\theta_k \sim \text{Beta}(\alpha/K, 1)$$



- Note that $P(z_{nk} = 1|\alpha) = E(\theta_k) = \frac{\alpha/K}{\alpha/K+1}$, so as K grows larger the matrix gets **sparser**.
- So if \mathbf{Z} is $N \times K$, the expected number of nonzero entries is $N\alpha/(1 + \alpha/K) < N\alpha$.
- Even in the $K \rightarrow \infty$ limit, the matrix is expected to have a finite number of non-zero entries.

Indian Buffet Process

We can **integrate out** θ , leaving:



$$\begin{aligned} P(\mathbf{Z}|\alpha) &= \int P(\mathbf{Z}|\theta)P(\theta|\alpha)d\theta \\ &= \prod_k \frac{\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(\frac{\alpha}{K})} \frac{\Gamma(1 + \frac{\alpha}{K})}{\Gamma(N + 1 + \frac{\alpha}{K})} \end{aligned}$$

The conditional feature assignments are:

$$\begin{aligned} P(z_{nk} = 1|\mathbf{z}_{-n,k}) &= \int_0^1 P(z_{nk}|\theta_k)p(\theta_k|\mathbf{z}_{-n,k}) d\theta_k \\ &= \frac{m_{-n,k} + \frac{\alpha}{K}}{N + \frac{\alpha}{K}}, \end{aligned}$$

where $\mathbf{z}_{-n,k}$ is the set of assignments of all objects, not including n , for feature k , and $m_{-n,k}$ is the number of objects having feature k , not including n .

We can take limit as $K \rightarrow \infty$.

“Rich get richer”, like in Chinese Restaurant Processes.

HMM vs iHMM

HMM is fully specified given

- K parameters
- K by K transition matrix

ϕ	ϕ_1	ϕ_2	ϕ_3	\dots	ϕ_K
π	π_{11}	π_{12}	\dots		
	π_{12}	\dots			
	\vdots				
					π_{KK}

HMM vs iHMM

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

ϕ	ϕ_1	ϕ_2	ϕ_3
π	π_{11}	π_{12}	...		
	π_{12}	...			
	⋮				
	⋮				
	⋮				

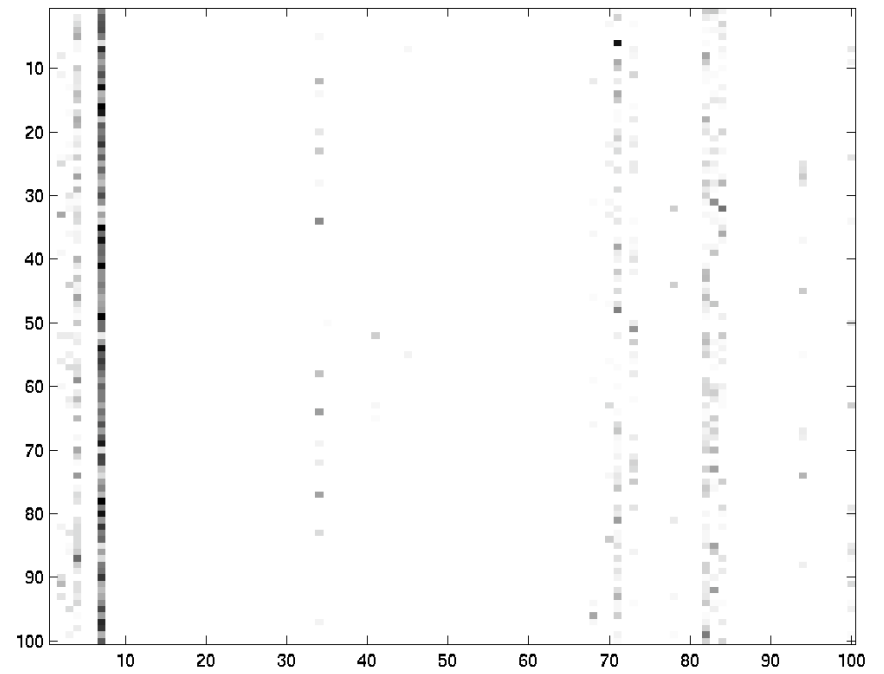
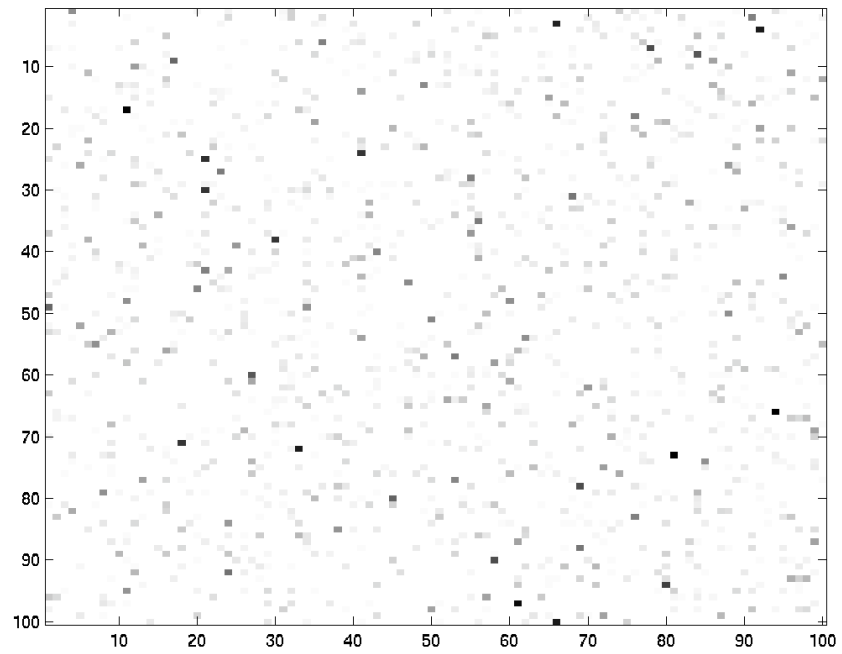
Infinite Hidden Markov Models

- Almost there: if H is continuous, all DP's will have different parameters
 - ➔ introduce a DP (G_0) between H and G_k

- Formally
$$G_0 \sim \text{DP}(\gamma, H)$$
$$G_k \sim \text{DP}(\alpha, G_0)$$

- Basic idea of two-level urn scheme to share information between states in (Beal, Ghahramani, and Rasmussen, 2002)
- Derived from Hierarchical Dirichlet Processes (Teh, Jordan, Beal & Blei, 2006)

iHMMs and HDPs



Inference and Learning

- Hidden Markov Model
 - Inference (= hidden states)
 - Dynamic Programming
 - Gibbs Sampling
 - Learning (= parameters)
 - Expectation Maximization
 - Gibbs Sampling
- Infinite Hidden Markov Model (so far)
 - Inference (= hidden states): Gibbs sampling
 - Learning (= parameters): Gibbs sampling
- This is unfortunate: Gibbs sampling for time series?!?