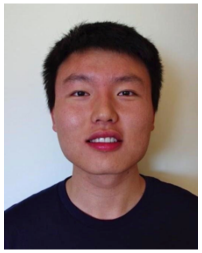
UNIVERSITY OF WASHINGTON

Variational Inference for Large-Scale and Streaming Sequential Data

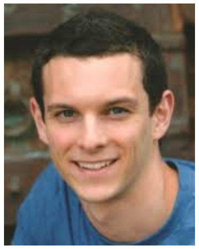
Emily Fox

Nick Foti

Alex Tank

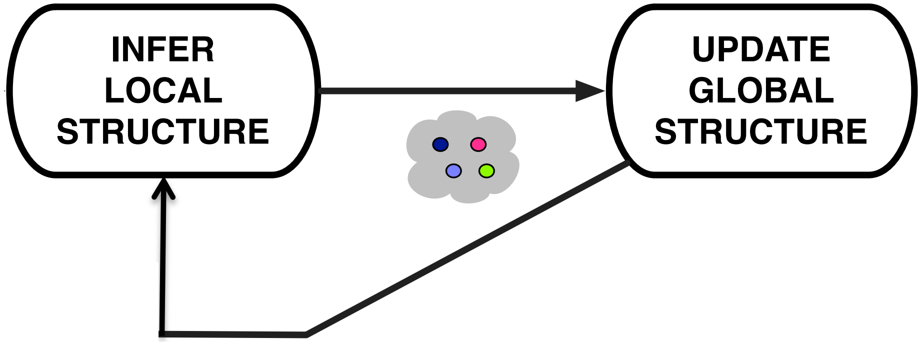


Jason Xu



Dillon Laird

Minibatch-Based Algorithms



INFER LOCAL STRUCTURE

UPDATE GLOBAL STRUCTURE

- Many ML/stat algorithms (e.g., gradient descent, Gibbs sampling,…) iterate between
  - operations involving all data
  - updating parameters

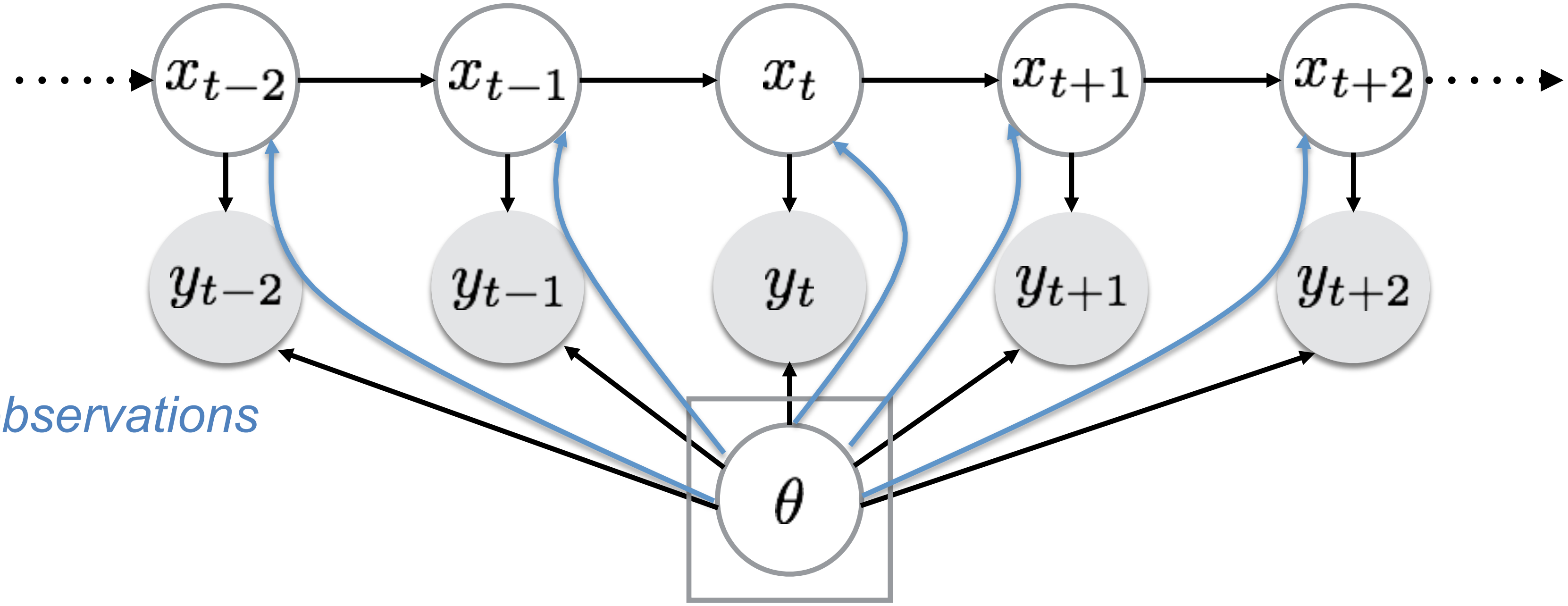
Not appropriate for dependent data

- Costly for large data / infeasible for streaming data

- Common approach for scalability:
  - *subsample data* → noisy operation
  - noisy update of parameters

Hidden Markov Models (HMMs)

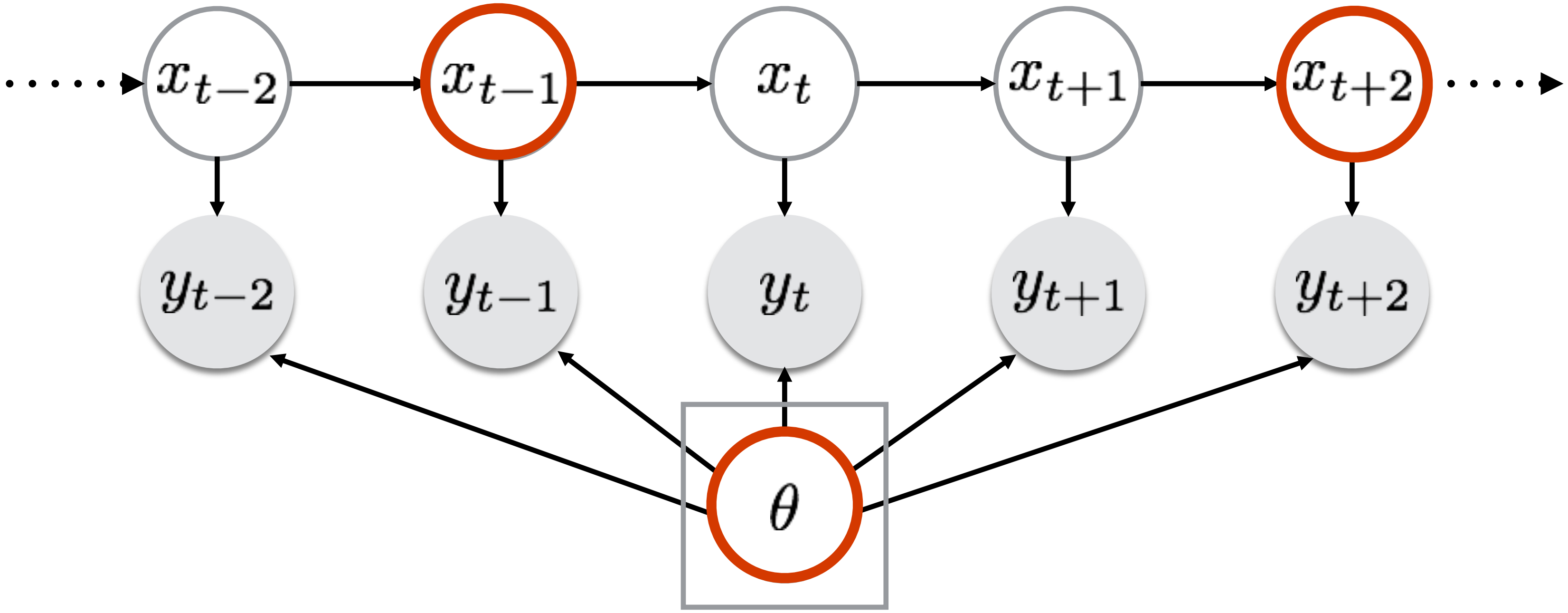
discrete state sequence



observations

transition probabilities,
observation parameters

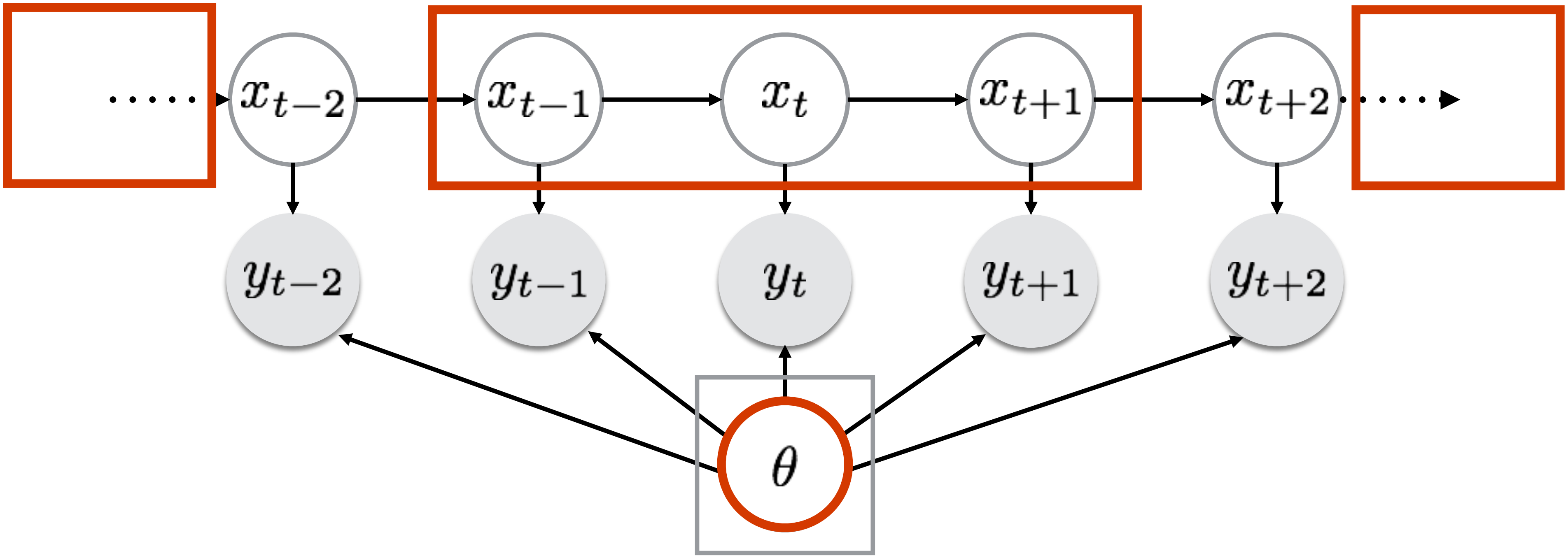
Minibatches for HMMs



- Why not just subsample observations independently?
- Cannot learn transition structure

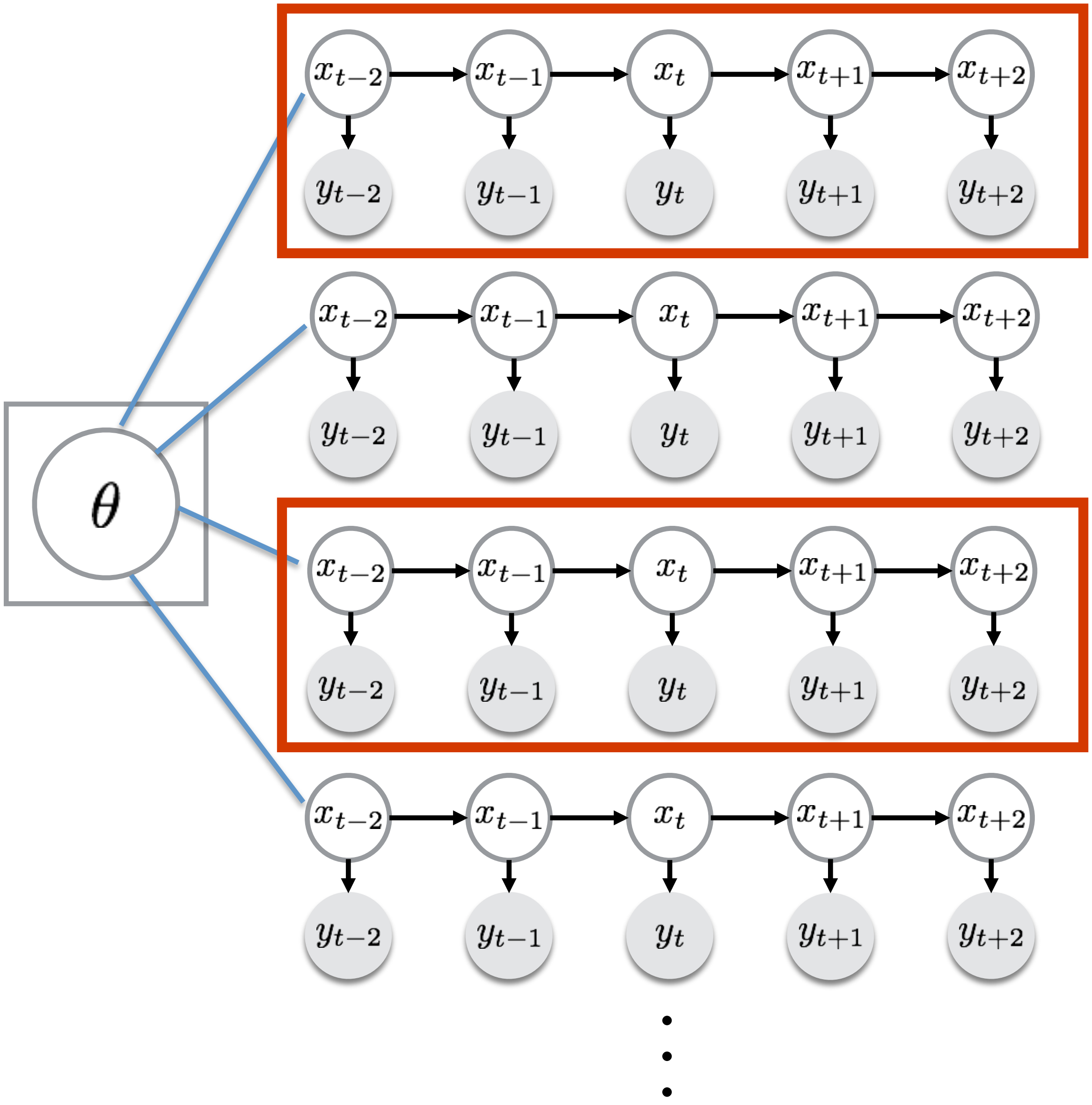
$$p(\mathbf{y}, \mathbf{x}, \theta) = p(\theta)\pi(x_1) \prod_{t=2}^{T} \boxed{p(x_t \mid x_{t-1}, \theta_A)} p(y_t \mid x_t, \theta_\phi)$$

Minibatches for HMMs



- How about sampling *subchain*? $x^S = (x_{t-L}, \dots, x_t, \dots, x_{t+L})$
- Do we just sever dependencies between subchains and analyze separately?

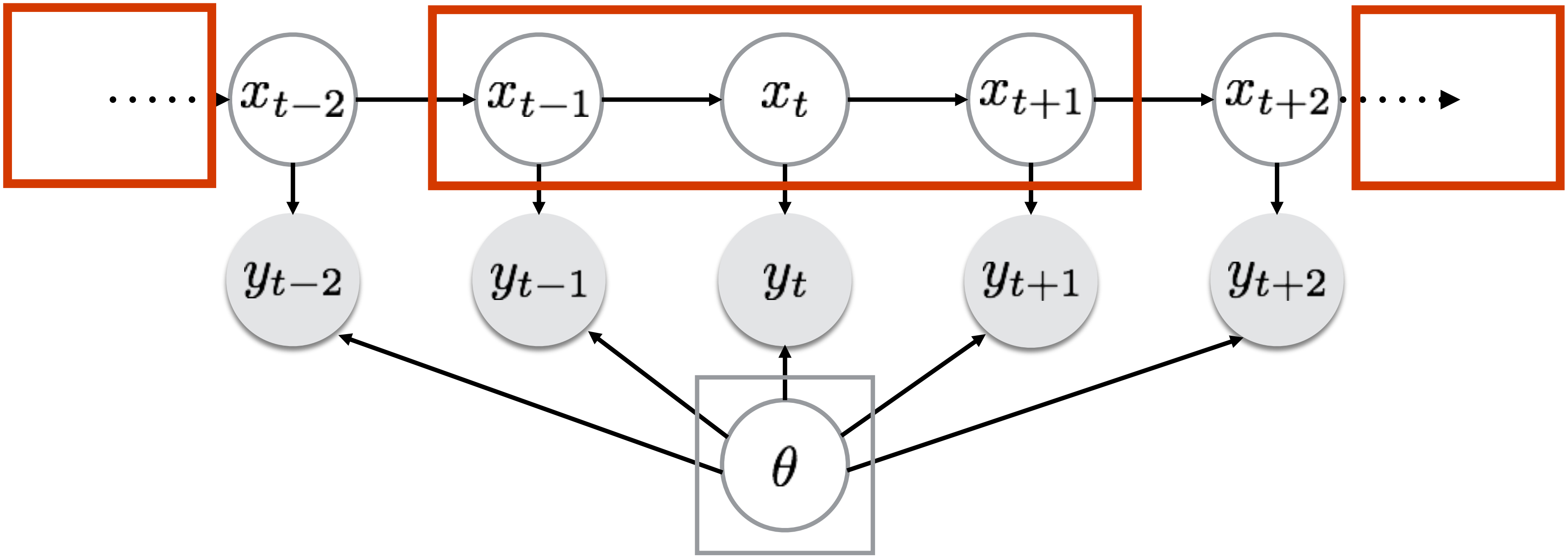
Large Collections of Short Chains



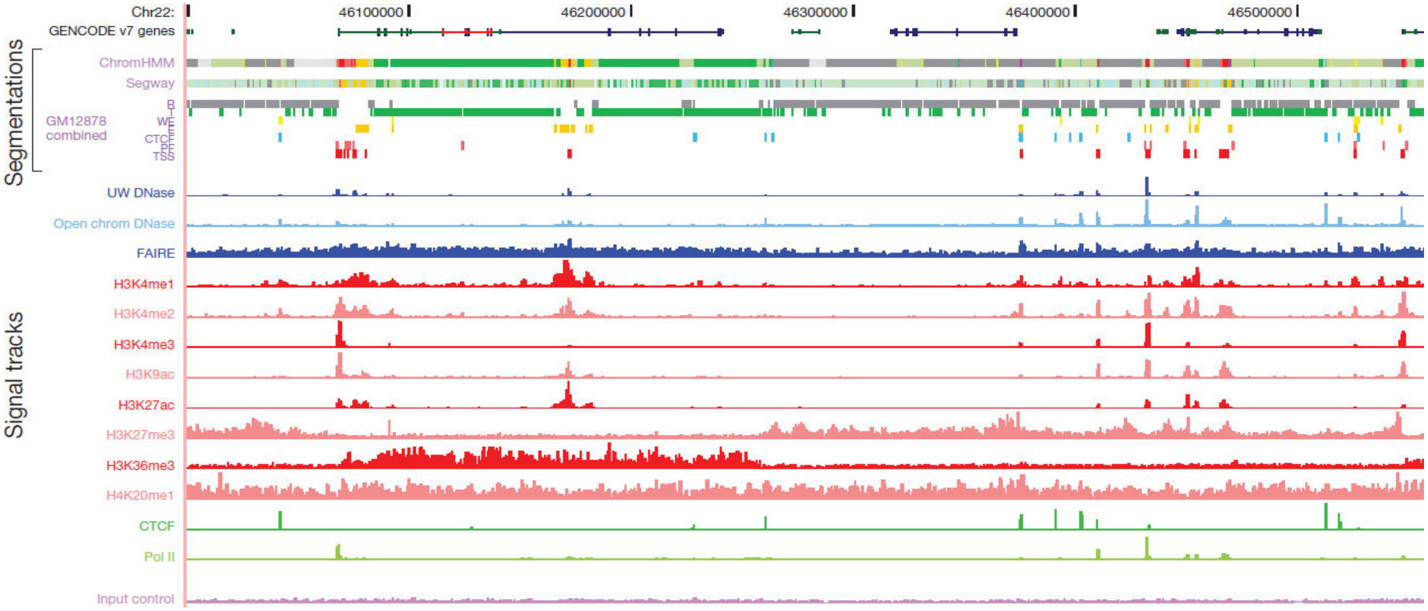
Johnson and Willsky,
ICML 2014

Hughes et al.,
NIPS 2015

One Long Chain



Human Chromatin Segmentation



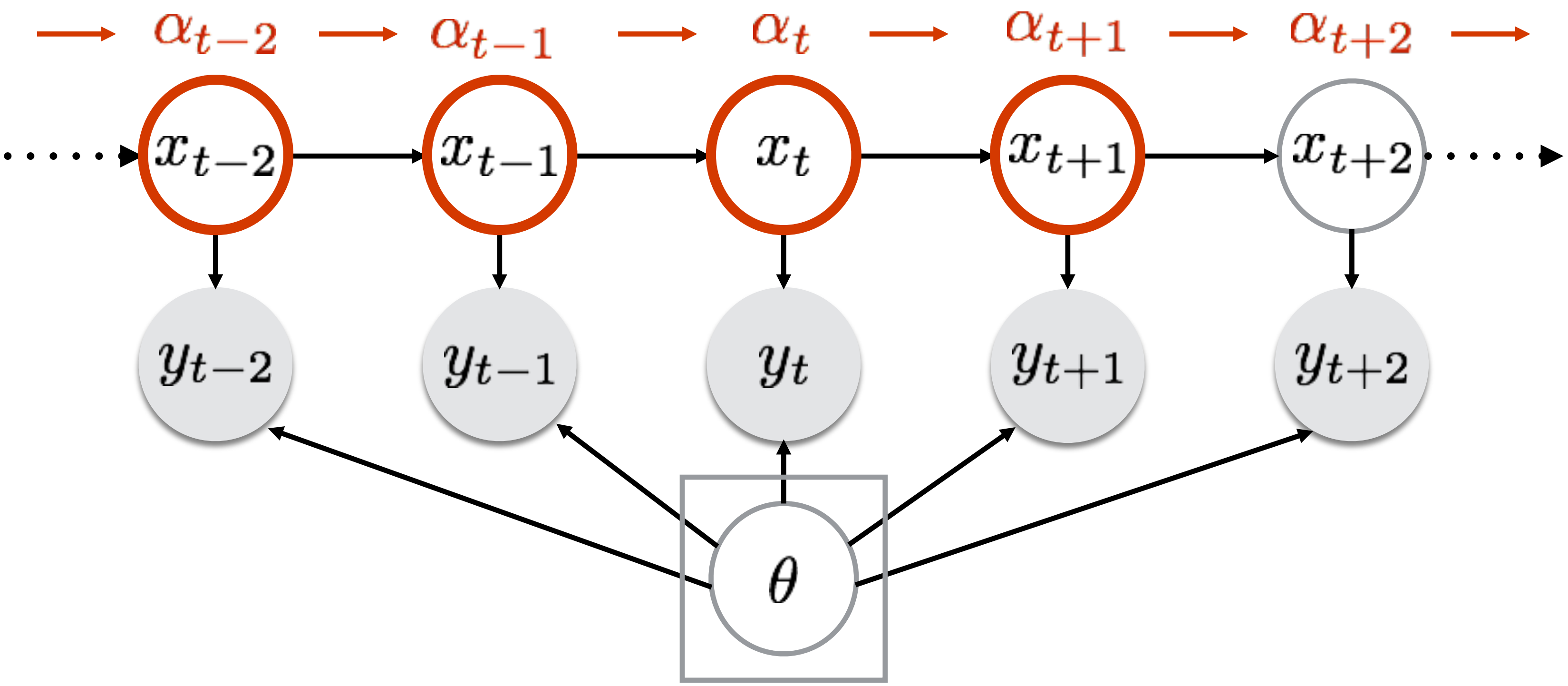
- Chromosome data set from the ENCODE project
  - ENcyclopedia Of DNA Elements
- 12 dimensional observations
- **Goal:** segment sequences

T = 250 million

BATCH LEARNING FOR HMMs

A quick review

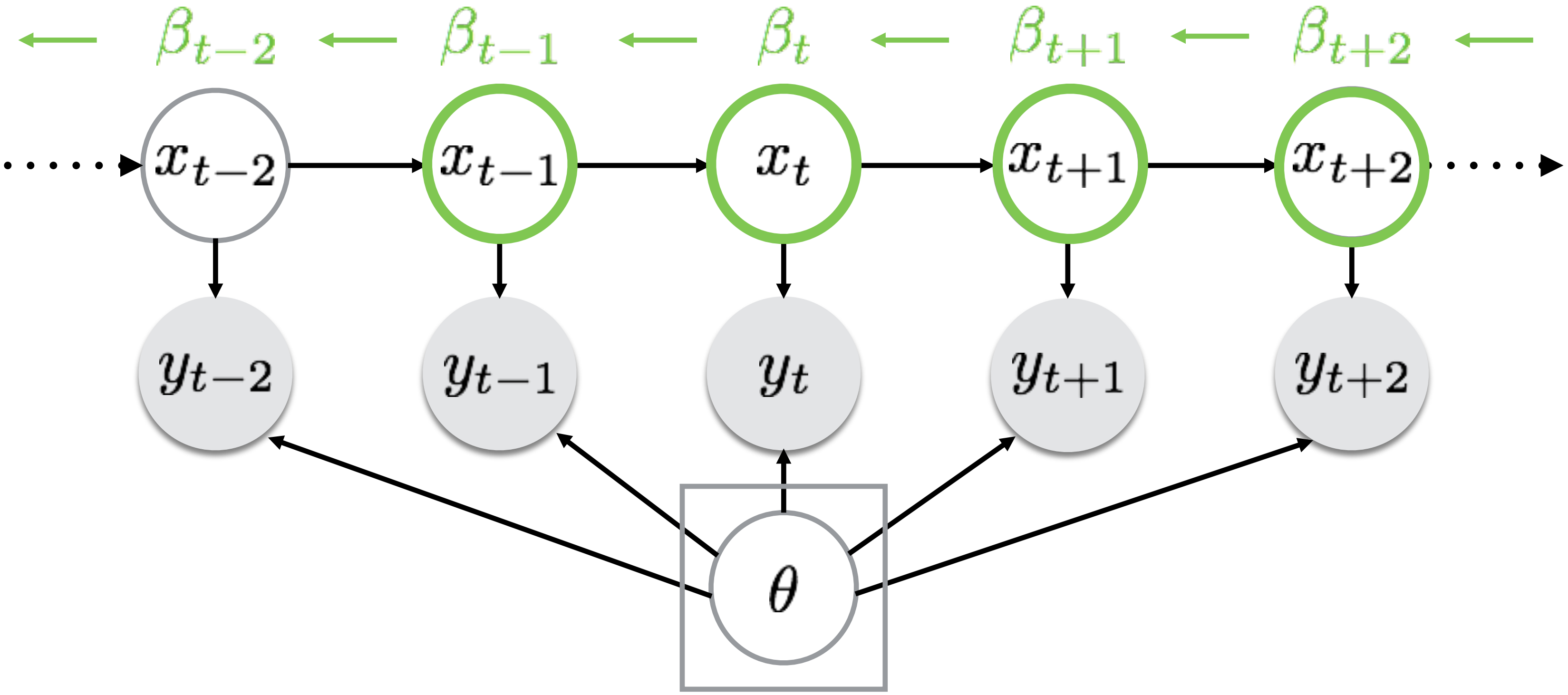
Batch Learning for HMMs



- Use current θ to form local state beliefs:
  - Propagate info forwards to form $\alpha_t = p(y_1, \dots, y_t, x_t)$

$$\alpha_{t+1,k} = p(y_{t+1} \mid x_{t+1} = k) \sum_{j=1}^{K} \alpha_{t,j} p(x_{t+1} = k \mid x_t = j)$$

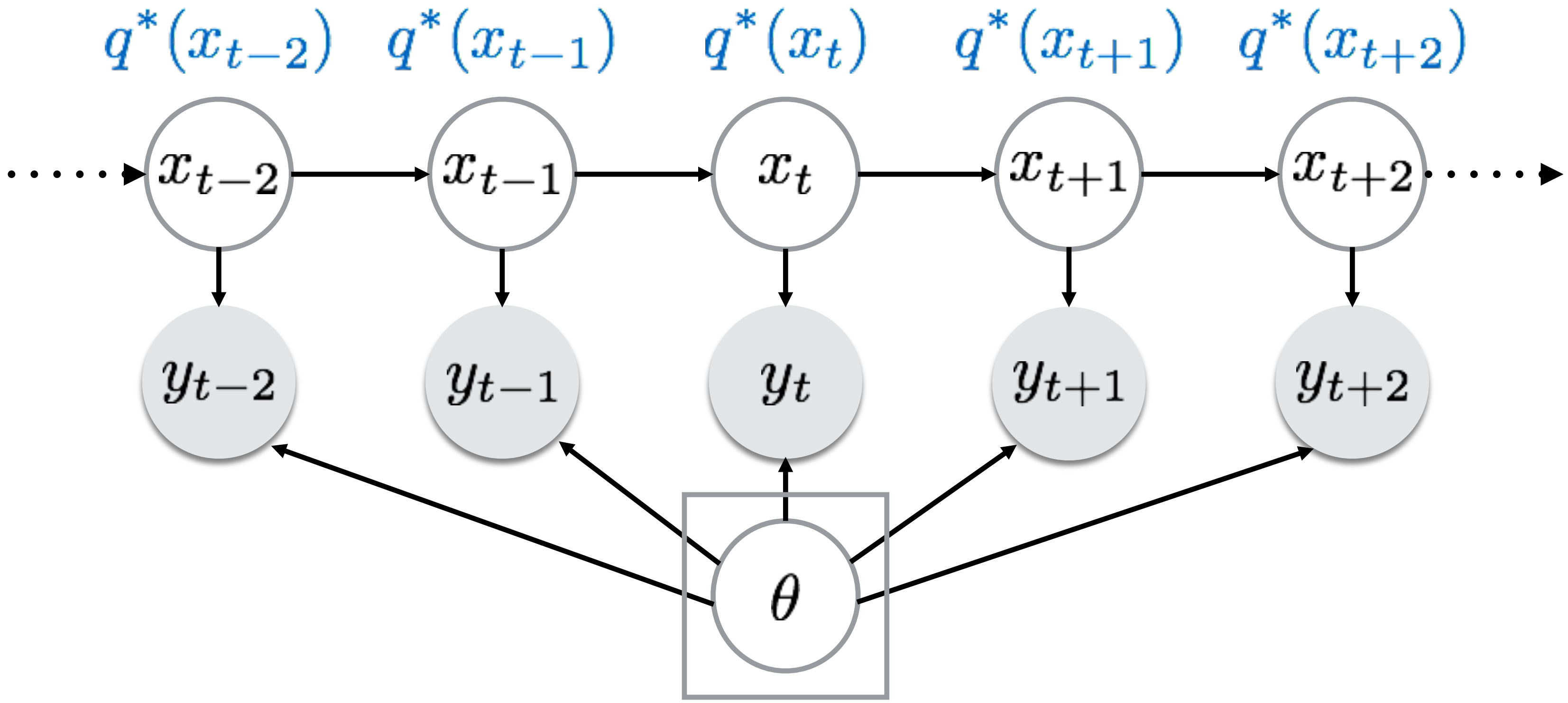
Batch Learning for HMMs



- Use current θ to form local state beliefs:
  - Propagate info backwards $\beta_t = p(y_{t+1}, \ldots, y_T \mid x_t)$

$$\beta_{t,k} = \sum_{j=1}^{K} p(y_{t+1} \mid x_{t+1} = j) p(x_{t+1} = j \mid x_t = k) \beta_{t+1,k}$$

Batch Learning for HMMs

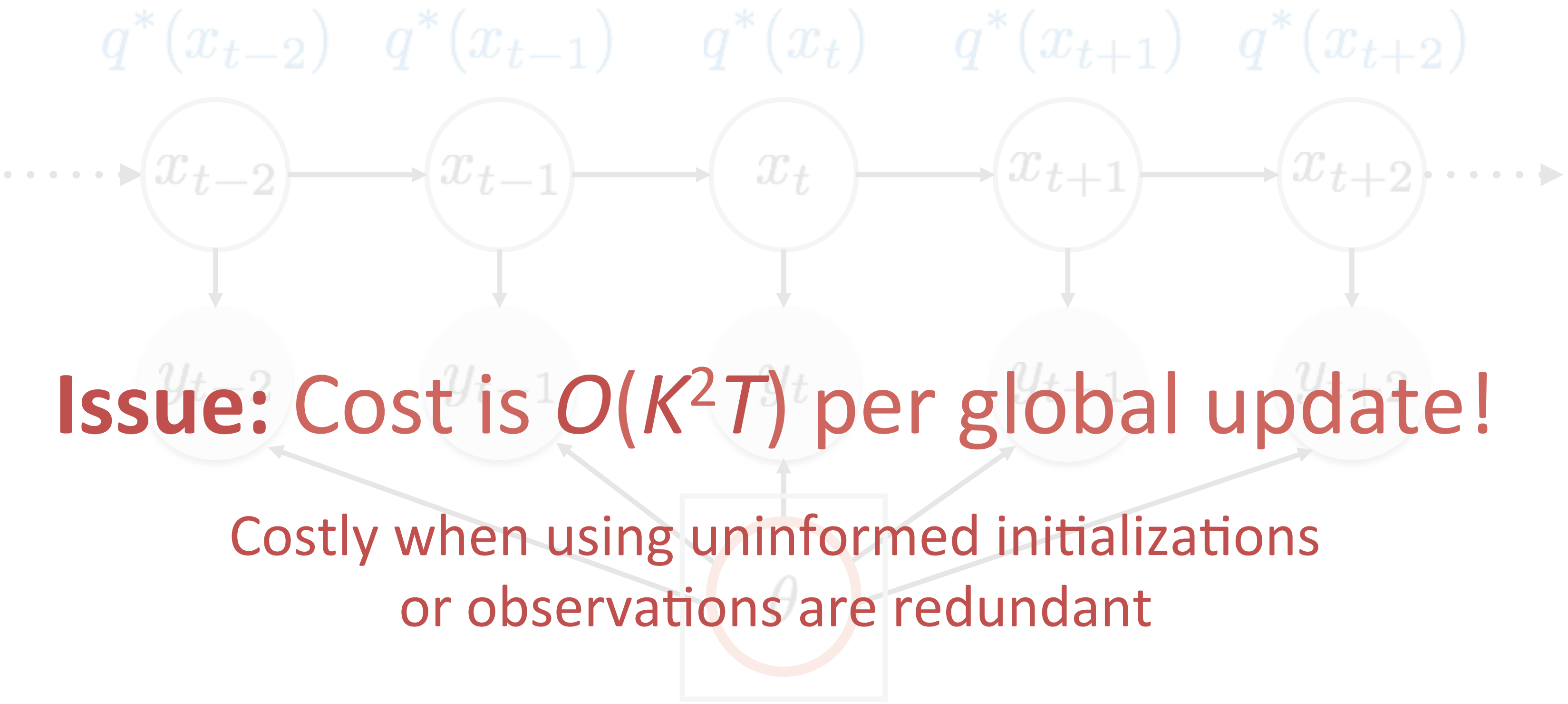


- Combine to form *smoothed* local state belief:

$$q^*(x_t) \propto \alpha_t \beta_t$$

$$p(x_t \mid y_1, \ldots, y_T, \theta)$$

Batch Learning for HMMs



Issue: Cost is $O(K^2T)$ per global update!

Costly when using uninformed initializations
or observations are redundant

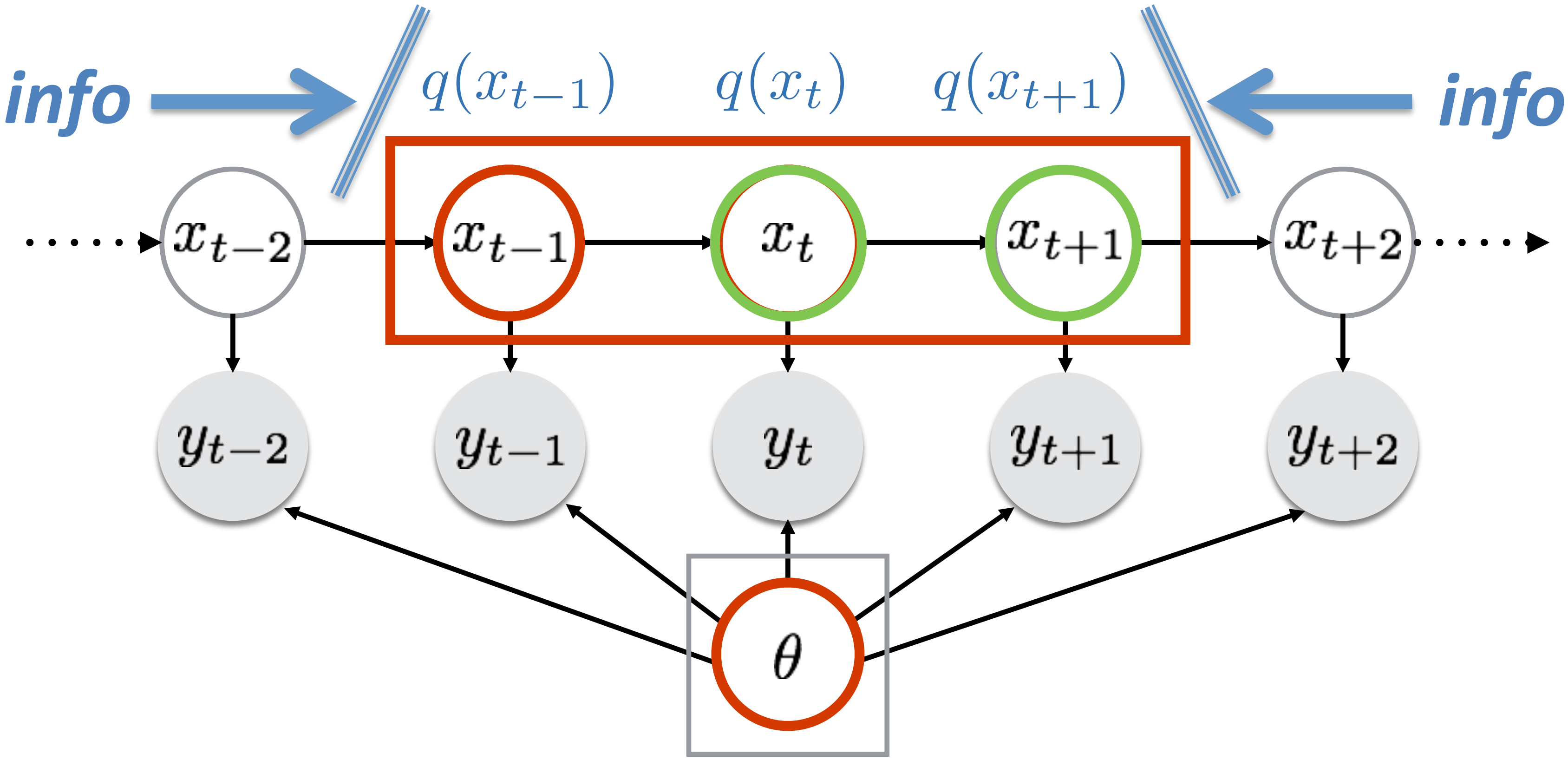
- Given local beliefs, update global parameter

T = 250 million

MINIBATCH LEARNING FOR HMMs?

Issues and solutions

Minibatch Inference for HMMs



- Form **local** beliefs $q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$ → perform **global** update

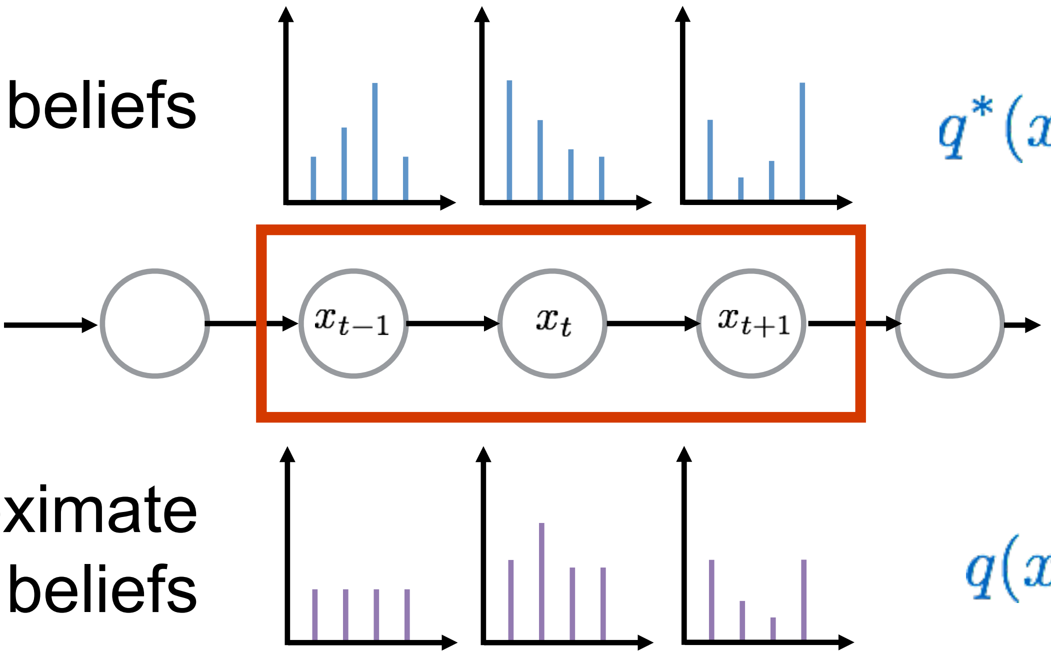
Local forward message

Local backward message

Harnessing Memory Decay

True beliefs

$q^*(x_t) \propto \alpha_t \beta_t$

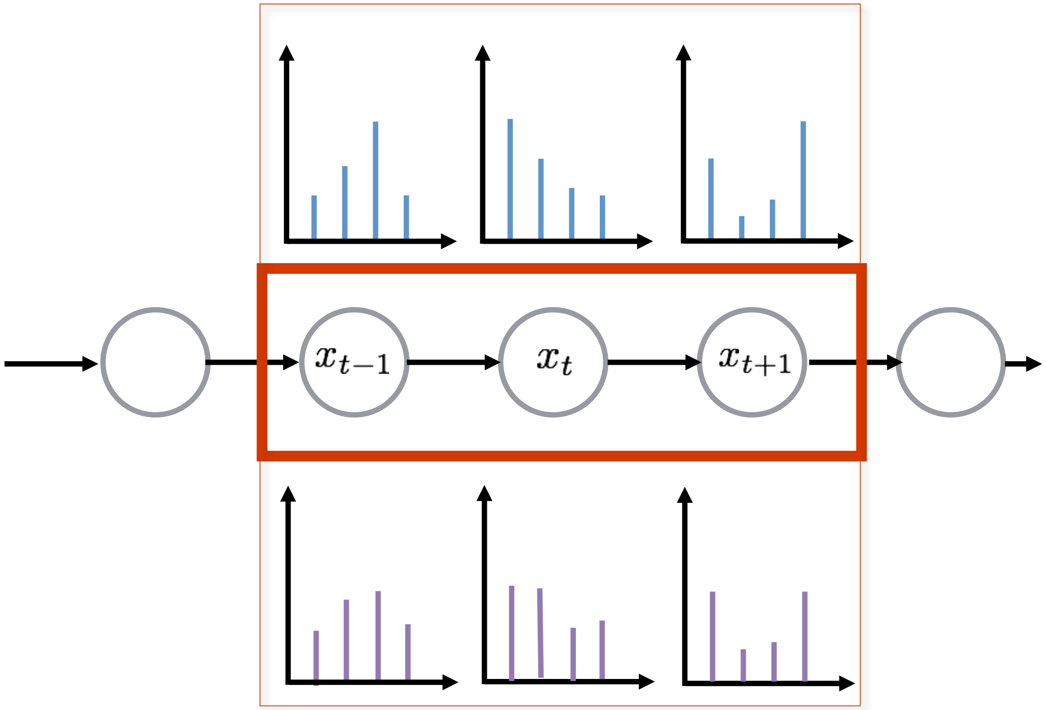


Approximate beliefs

$q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$

Do we expect x_t to influence $x_{t+1,000,000}$?

Buffering Subchains



$$q^*(x_t) \propto \alpha_t \beta_t$$

$$q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$$

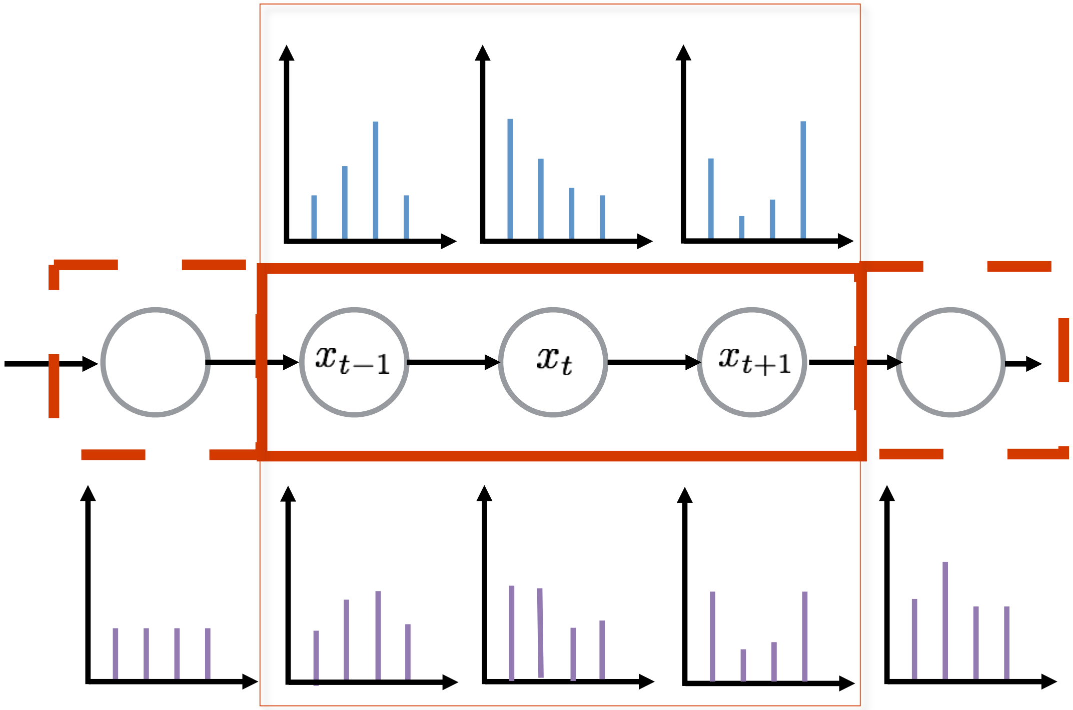
Check that subchain marginals are approximated well:

$$\max_{i \in S} ||q(x_i) - q^*(x_i)|| < \epsilon \quad ?$$

Local subchain marginal

Full data marginal

Buffering Subchains



$$q^*(x_t) \propto \alpha_t \beta_t$$

$$q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$$

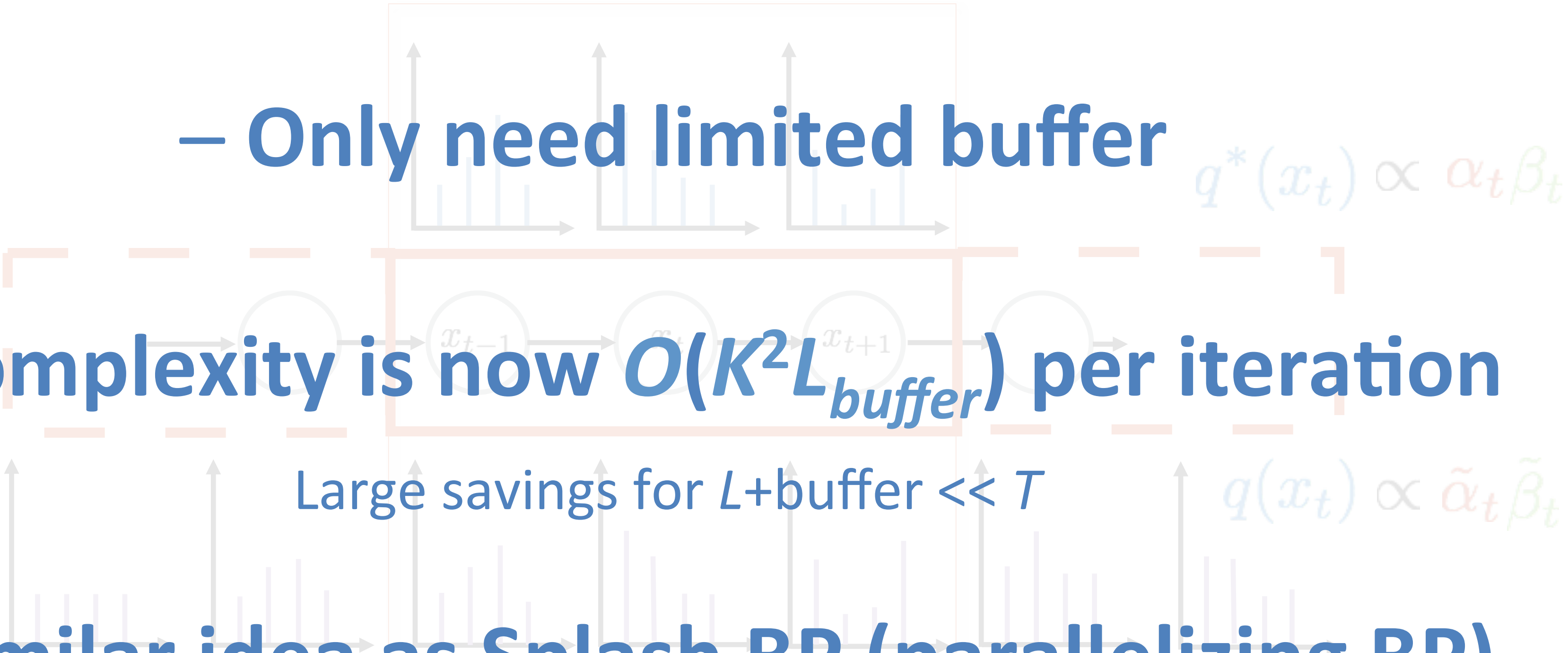
Check that subchain marginals are approximated well:

$$\max_{i \in S} ||q(x_i) - q^*(x_i)|| < \epsilon \quad ?$$

Local subchain marginal

Full data marginal

Buffering Subchains


x_{t-1}
x_t
x_{t+1}
$q^*(x_t) \propto \alpha_t \beta_t$
$q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$

– **Only need limited buffer**

– **Complexity is now $O(K^2 L_{buffer})$ per iteration**

Large savings for *L*+buffer << *T*

– **Similar idea as Splash BP (parallelizing BP)**

[Gonzalez, et. al. 2009]

Check that subchain marginals are approximated well:

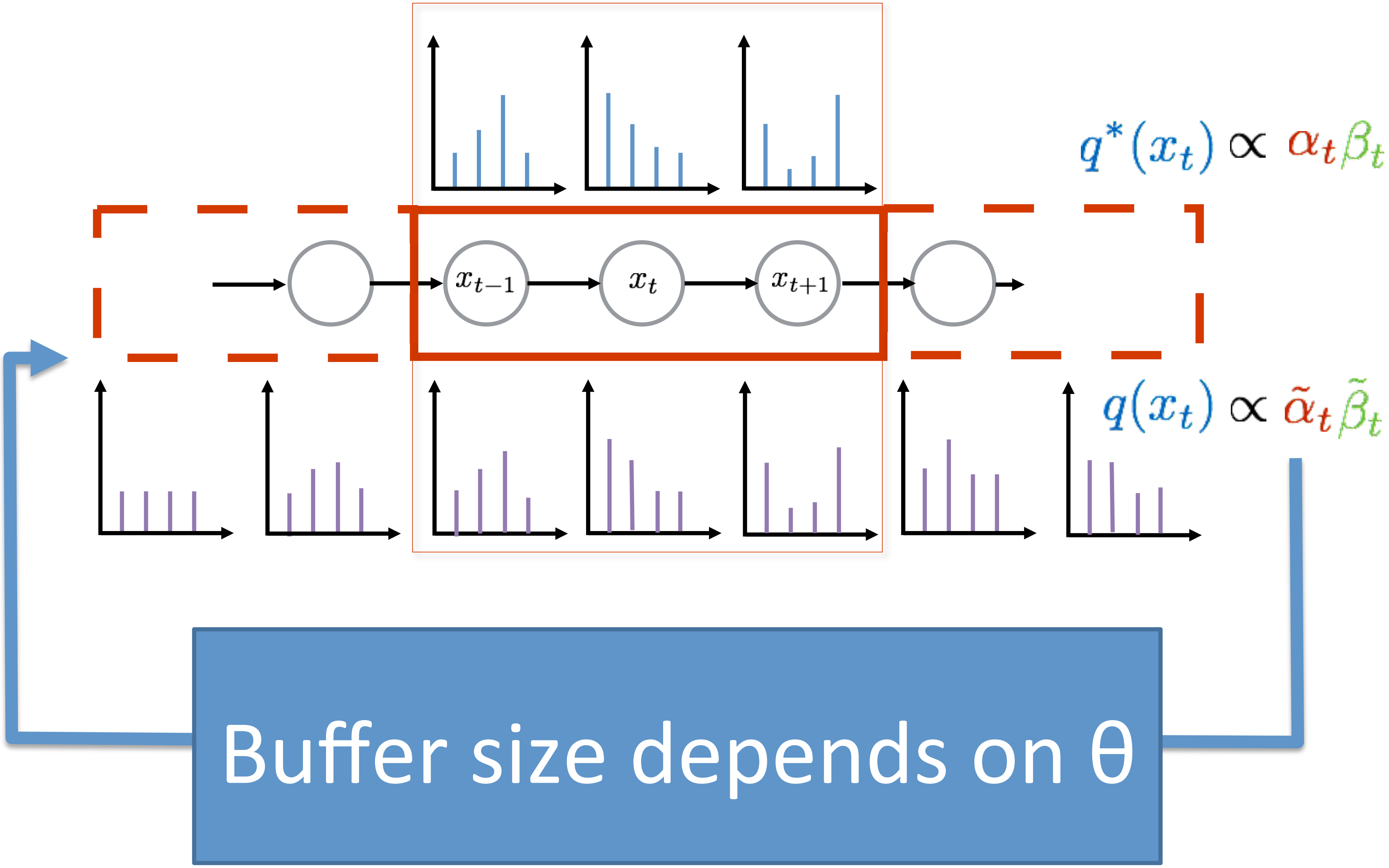
$\max_{i \in S} ||q(x_i) - q^*(x_i)|| < \epsilon$ ✓

But, uncertain parameter setting here

Local subchain marginal

Full data marginal

Buffering for Learning


$q^*(x_t) \propto \alpha_t \beta_t$
x_{t-1}
x_t
x_{t+1}
$q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$

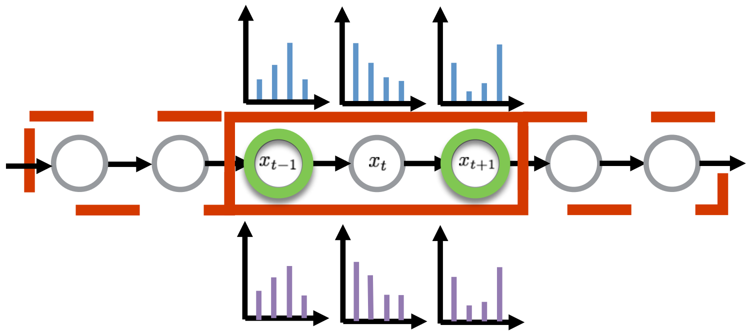
Buffer size depends on θ

Buffering in Practice

- We do not actually know the true marginals
- Monitor changes in approximate subchain beliefs:

$$\max_{i \in S} \left|\left| q(x_i)^{\text{new}} - q(x_i)^{\text{old}} \right|\right| < \epsilon$$

- Chain structuring implies that only endpoints must be checked



- During buffer expansions, forward-backward passes can reuse computations of previous buffer

A CASE STUDY: SVI-HMM

Minibatch-based variational Bayes for HMMs

Variational Bayes (VB)

- Approximate posterior with variational distribution

$$p(\underset{\text{latent variables}}{\underset{\uparrow}{x}}, \overset{\text{parameters}}{\overset{\downarrow}{\theta}} | \underset{\text{observations}}{\underset{\uparrow}{y}}) = \frac{p(y|x,\theta)p(x,\theta)}{p(y)} \approx q(x,\theta)$$

- Minimize $\mathrm{KL}(q||p) \leftrightarrow$ maximize “ELBO”:

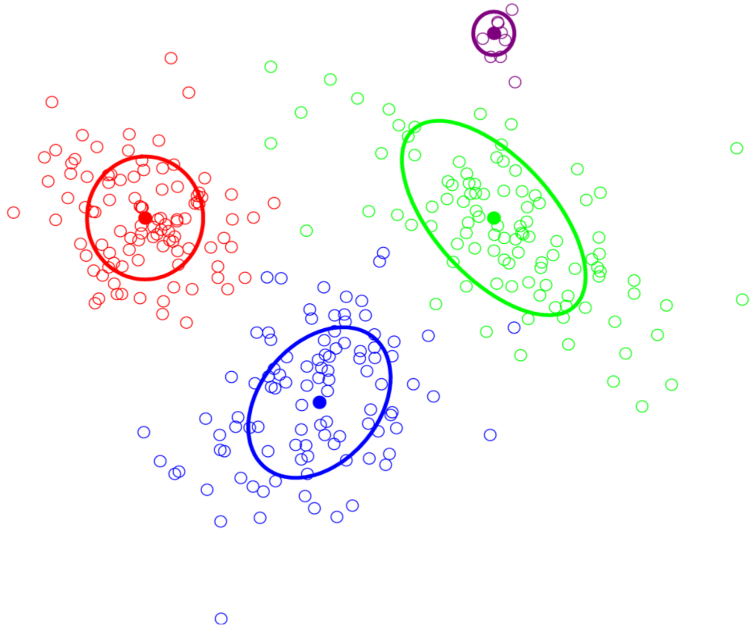
$$\mathcal{L}(q) = \mathbb{E}_q[\log p(y,x,\theta)] - \mathbb{E}_q[\log q(x,\theta)] \leq \log p(y)$$

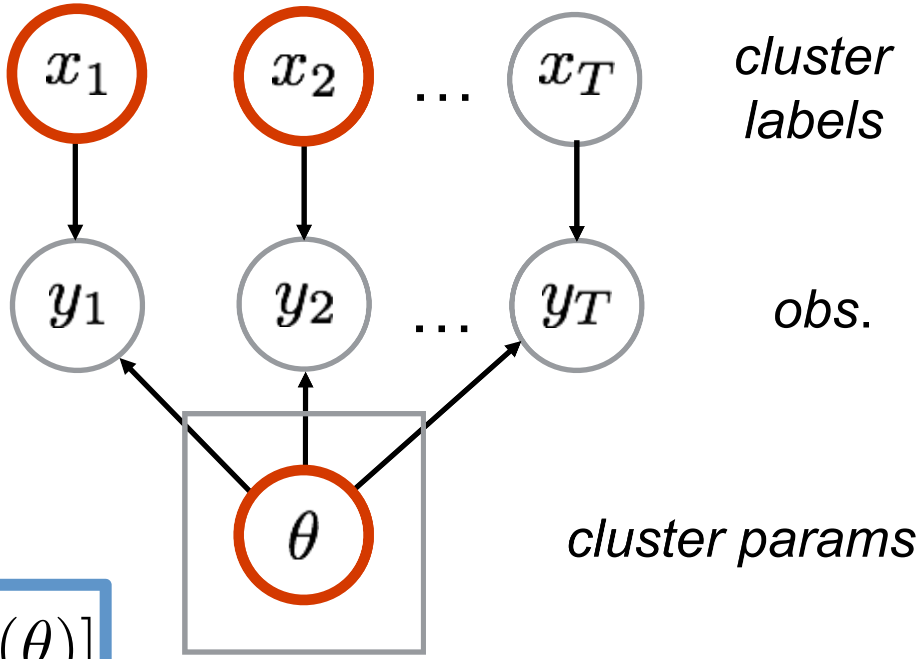
- Common to make mean-field assumption:

$$q(x,\theta) = q(x)q(\theta)$$

VB Example: Mixture Model

Maximize ELBO with *coordinate-ascent* $\frac{\partial \mathcal{L}}{\partial q(\mathbf{x})} = 0 \leftrightarrow \frac{\partial \mathcal{L}}{\partial q(\theta)} = 0$

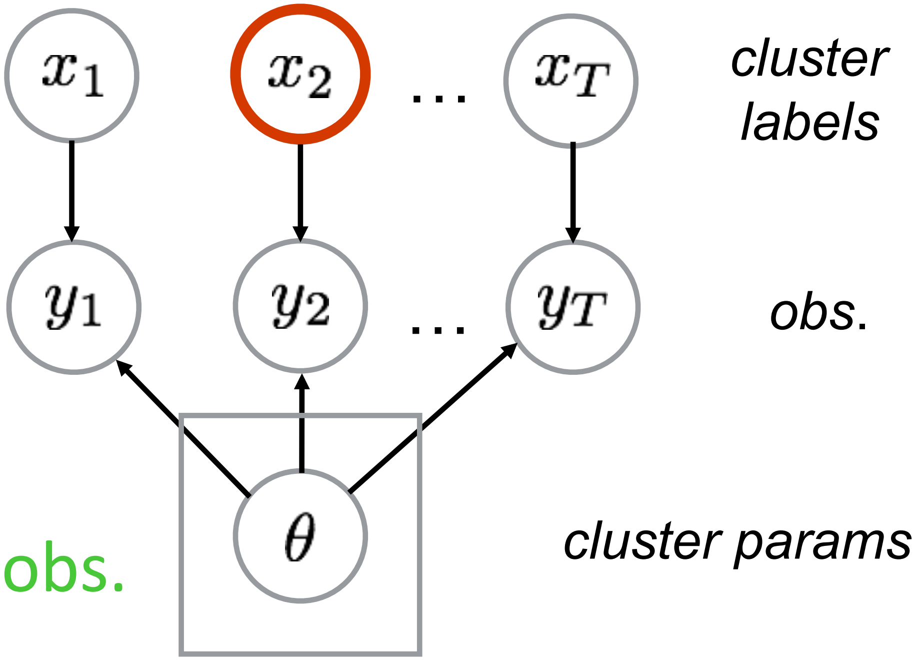




$$\mathcal{L} = E_{q(\theta)}\left[\ln p(\theta)\right] - E_{q(\theta)}\left[\ln q(\theta)\right] + \sum_{i=1}^{T} E_{q(x_i)}\left[\ln p(y_i, x_i|\theta)\right] - E_{q(x_i)}\left[\ln q(x_i)\right]$$

SVI Example: Mixture Model

For scalability, stochastic variational inference (SVI) replaces global coordinate step with *stochastic gradient* step [Hoffman, et. al. 2013]



1. Sample observation uniformly at random

$$x^S \sim \mathrm{Unif}(x_1, \ldots, x_T)$$

2. Form noisy, unbiased ELBO:

As if we saw obs. *T* times

$$\mathcal{L}^s = E_{q(\theta)}\left[\ln p(\theta)\right] - E_{q(\theta)}\left[\ln q(\theta)\right] \\ + T \cdot \left(E_{q(x_s)}\left[\ln p(y_s, x_s|\theta)\right] - E_{q(x_s)}\left[\ln q(x_s)\right]\right)$$

SVI Example: Mixture Model

3. Take standard coordinate step for x^S

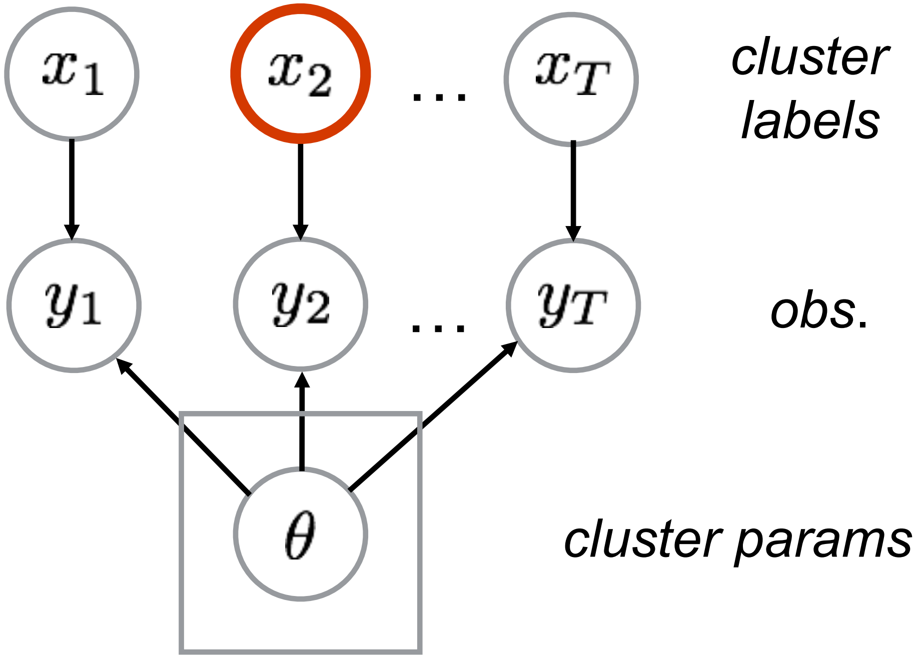
$$\frac{\partial \mathcal{L}^s}{\partial q(x_s)} = 0$$

4. Take stochastic natural gradient step for θ

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \rho_t \tilde{\nabla}_{\mathbf{w}} \mathcal{L}^S$$

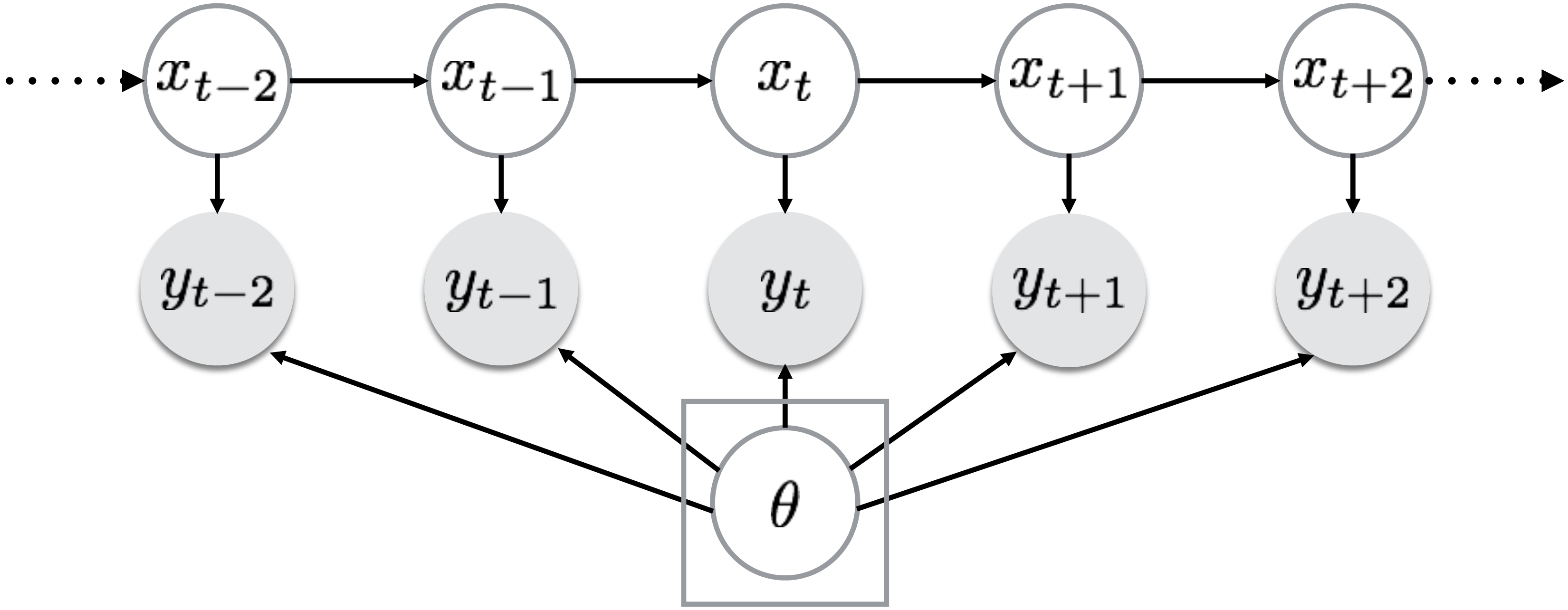
Hyperparams for q(θ)

5. Iterate



$$\mathcal{L}^s = E_{q(\theta)}\left[\ln p(\theta)\right] - E_{q(\theta)}\left[\ln q(\theta)\right] \\ + T \cdot \left(E_{q(x_s)}\left[\ln p(y_s, x_s|\theta)\right] - E_{q(x_s)}\left[\ln q(x_s)\right]\right)$$

Variational Inference for HMMs



Use structured mean-field approximation:

$$p(x_1, x_2, \ldots, x_T, \theta \mid y_1, y_2 \ldots, y_T) \approx q(x_1, x_2, \ldots, x_T) q(\theta)$$

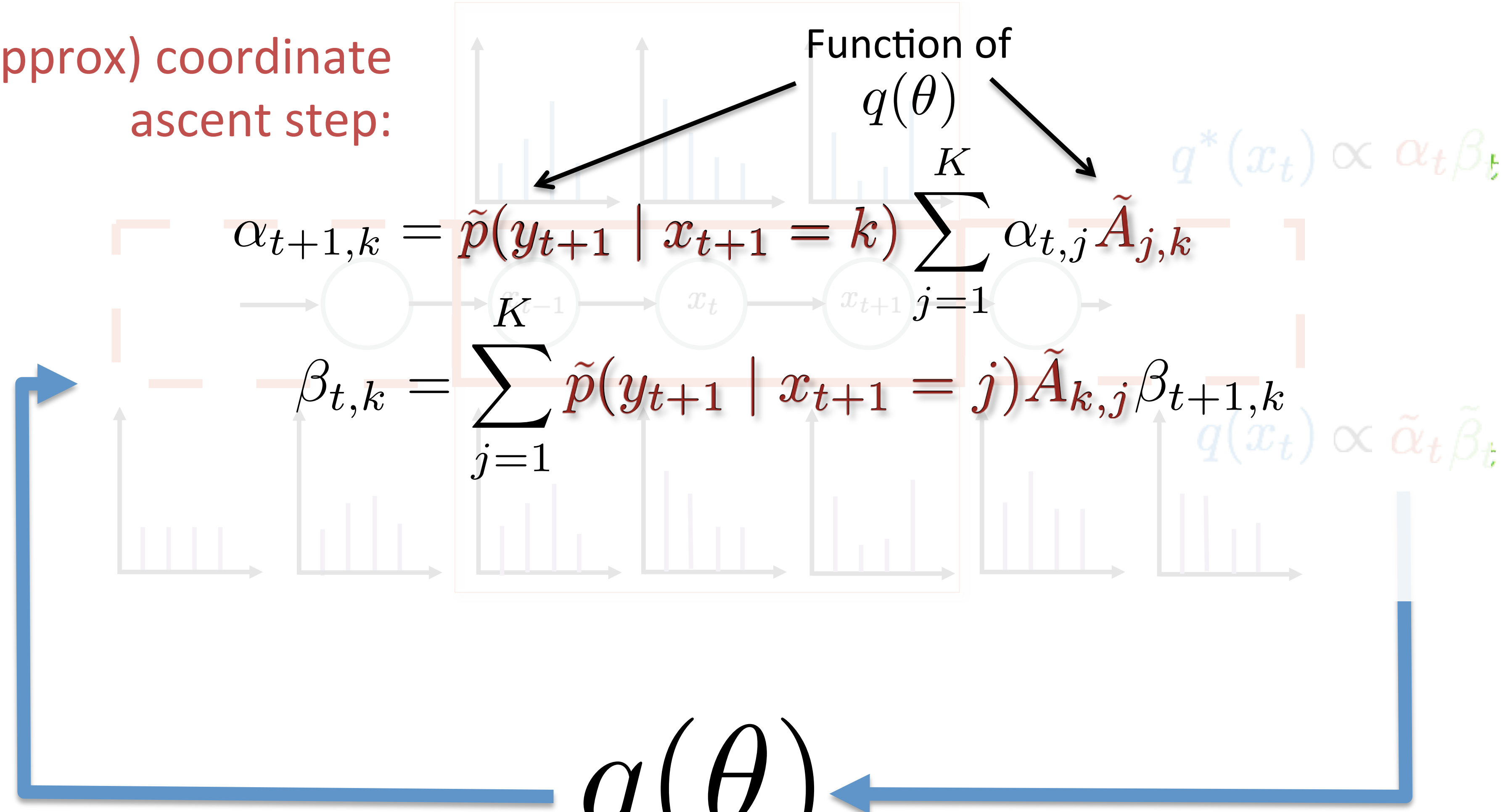
SVI for HMMs

(Approx) coordinate ascent step:

Function of $q(\theta)$

$$\alpha_{t+1,k} = \tilde{p}(y_{t+1} \mid x_{t+1} = k) \sum_{j=1}^{K} \alpha_{t,j} \tilde{A}_{j,k}$$

$$\beta_{t,k} = \sum_{j=1}^{K} \tilde{p}(y_{t+1} \mid x_{t+1} = j) \tilde{A}_{k,j} \beta_{t+1,k}$$


$q^*(x_t) \propto \alpha_t \beta_t$
x_{t-1}
x_t
x_{t+1}
$q(x_t) \propto \tilde{\alpha}_t \tilde{\beta}_t$

$$q(\theta)$$

Stochastic natural gradient step:

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \rho_t \tilde{\nabla}_{\mathbf{w}} \mathcal{L}^S$$

Function of $q(\mathbf{x})$

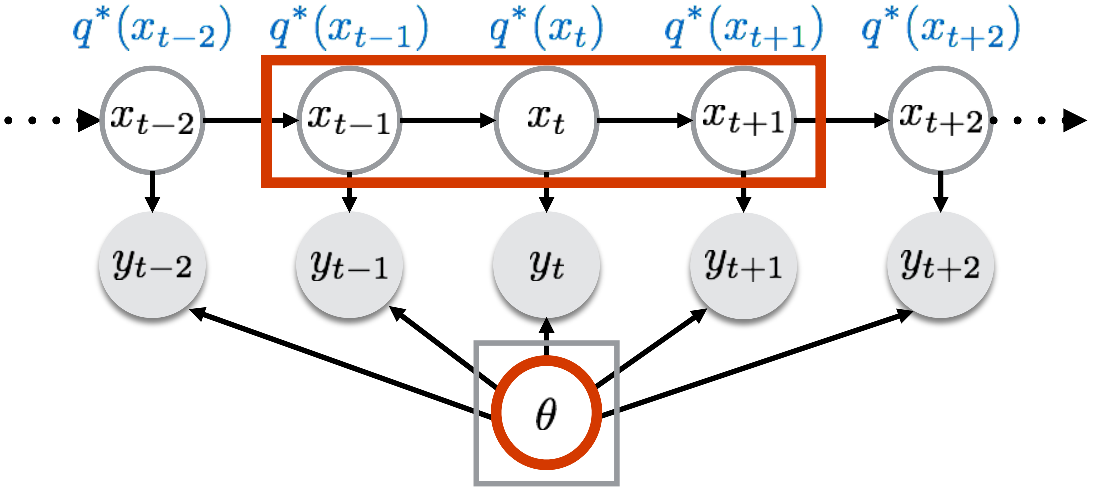
Foti, Xu, Laird, Fox, *NIPS 2014*

Differences from i.i.d. Case

- Minibatches are *correlated*
  - Data in one is not independent of data in another
- Minibatch marginals ≠ batch marginals
  - Impact of latent chain
  - Mitigated by buffering

Correlated Minibatches

- Pretend we have exact local distribution $q^*(x^S)$



As if we had run batch forward-backward

- Typical arguments for convergence to local mode rely on *unbiased* + *independent* noisy gradients [c.f., Bottou 1998, Hoffman 2013]
  – Our SGs are *dependent* since subchains are correlated
- Using [Polyak and Tsypkin 1973], unbiasedness suffices for convergence of $\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \rho_t \tilde{\nabla}_{\mathbf{w}} \mathcal{L}^S$

Effect of Approximated Marginals

SVI-HMM iterates:

buffer minibatches to **approx *q(x)*** $\longleftrightarrow$ **update *q(Θ)***

coordinate ascent step

stochastic (natural) gradient step

For ϵ sufficiently small (sufficiently long buffer)

- Approximate marginals “close enough” to true marginals
- Noisy gradient in same half-plane as true gradient

$$\Downarrow$$

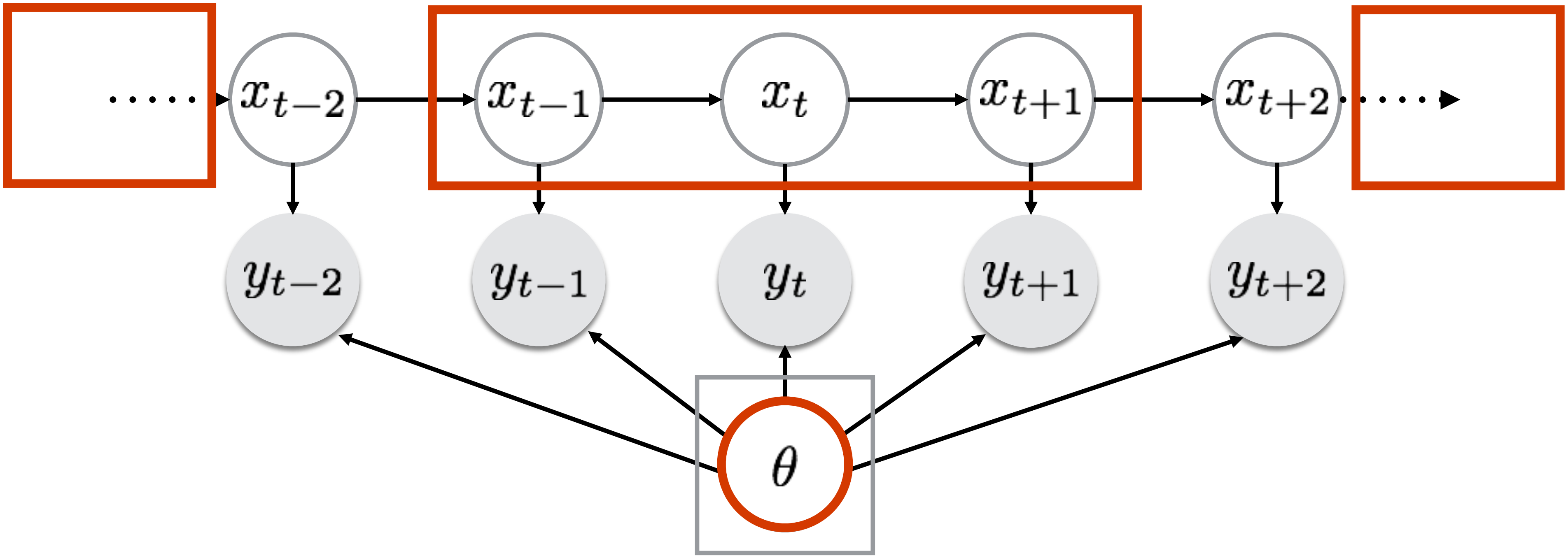
iterative algorithm converges to local mode of ELBO

Foti, Xu, Laird, Fox, *NIPS 2014*

Experiments

- Synthetic data:
  - **Diagonally Dominant**: Long memory chain with large self-transitions
  - **Reversed Cycles**: Two overlapping cycles with opposite directions
- **Human chromatin application**

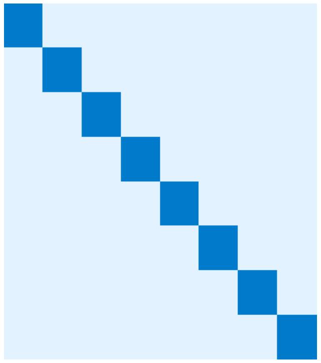
Minibatch of Subchains

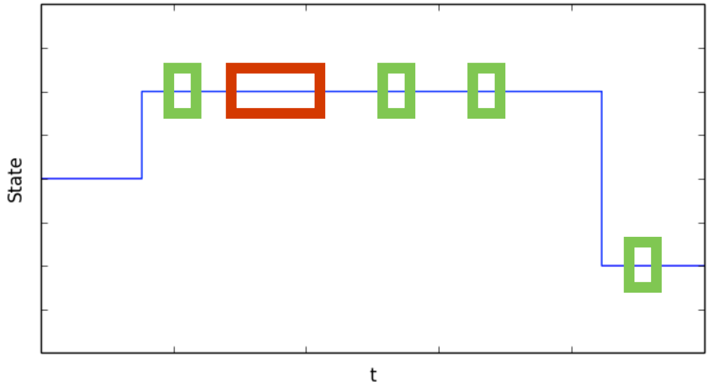


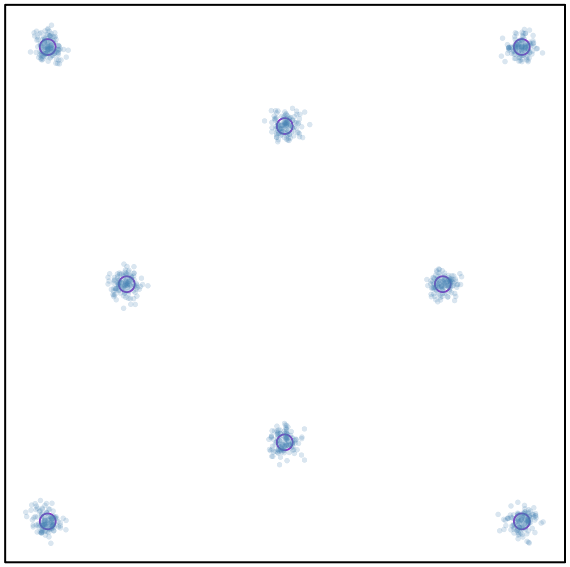
Minibatch consists of *M* subchains each of length *L*

Diagonally Dominant

- 8 latent states
- 2d Gaussian emissions





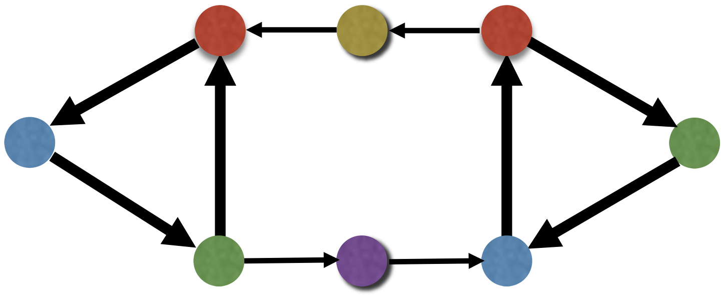


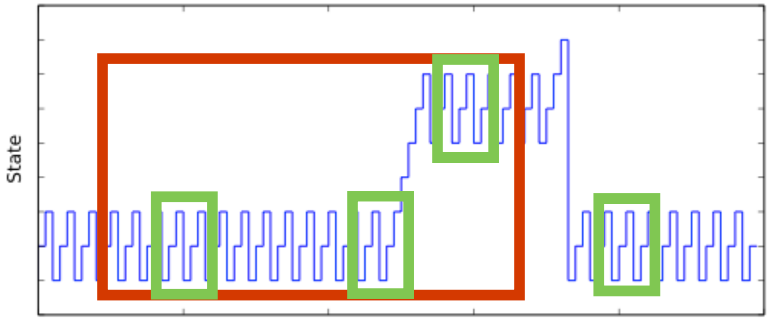
- *High auto-correlation*
  ➔ few long subchains converge slowly (small *M*, large *L*)

- *Emissions identifiable*
  ➔ many small subchains perform better (large *M*, small *L*)

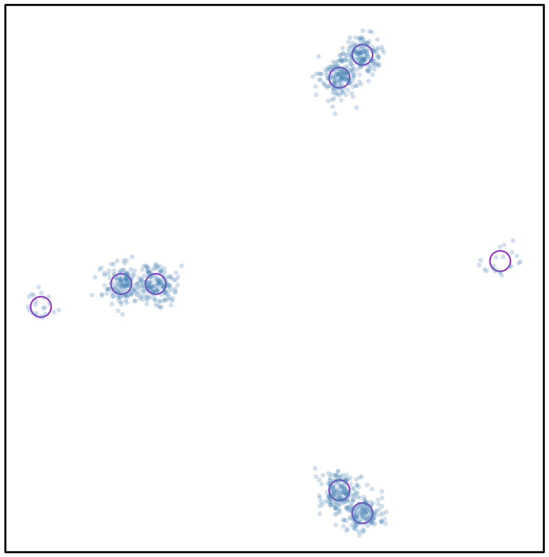
Reversed Cycles

- 8 latent states
- 2d Gaussian emissions



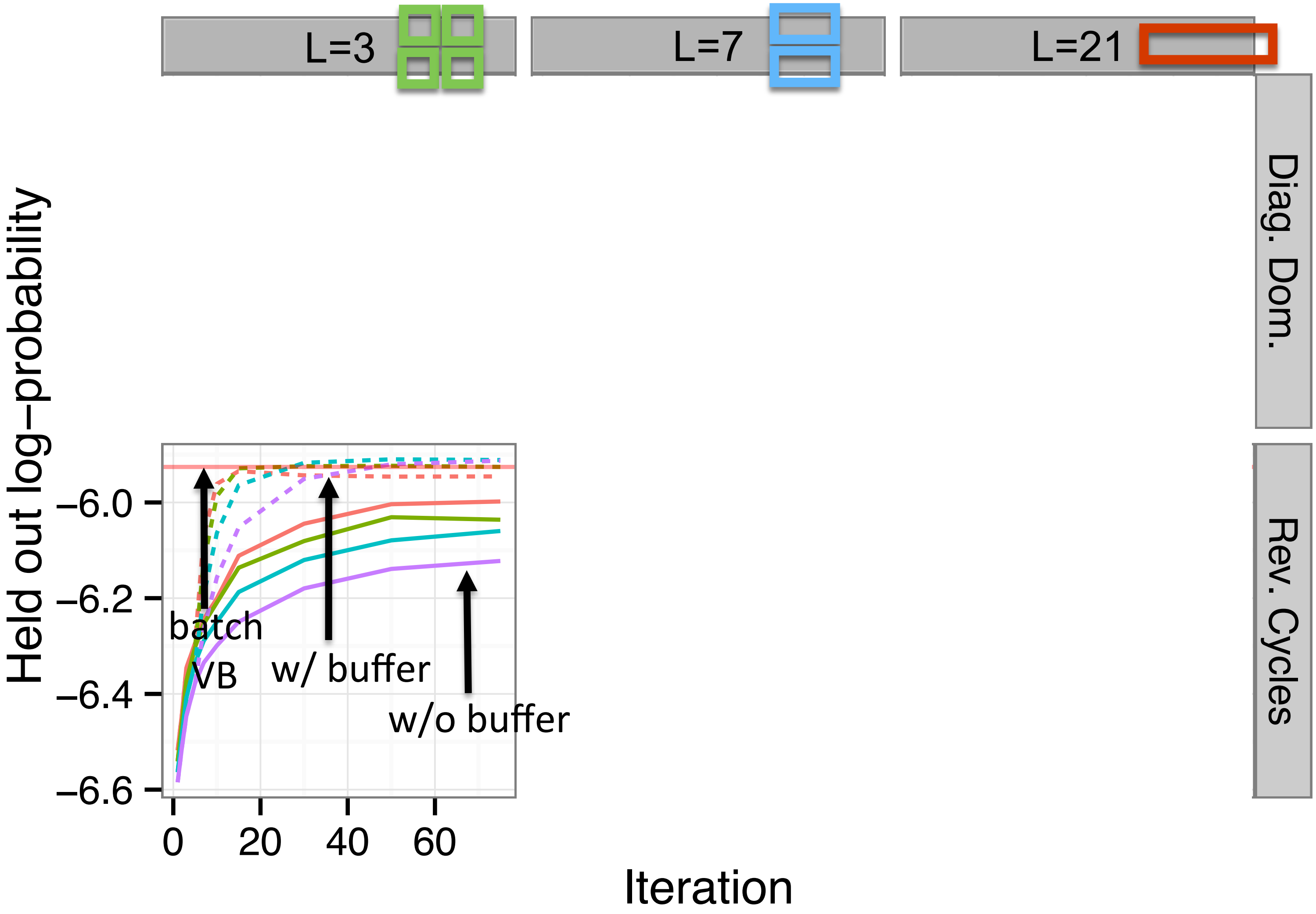


- *Emission distributions overlap*
- *Direction* of cycles important to identify states
  - Singleton observations insufficient
  - Without buffering, need *L* > 3 to learn effectively



- Longer subchains more likely to capture structure

Subchain Buffering



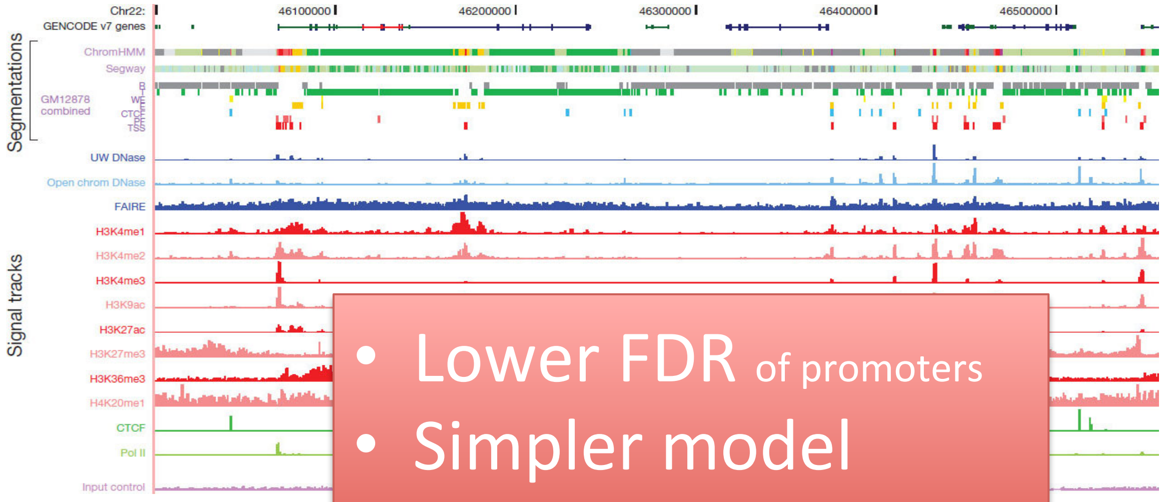
batch VB

w/ buffer

w/o buffer

Human Chromatin Segmentation

- Chromosome data from ENCODE project
- 12 dimensional observations
- **Goal:** segment sequences
- **T = 250 million**



- Lower FDR of promoters
- Simpler model
- Uses all of the data

- [Hoffman et. al. 2012] used **dynamic Bayesian network**
  - Broke sequence into pieces to perform inference via EM
  - Severs long-range dependencies

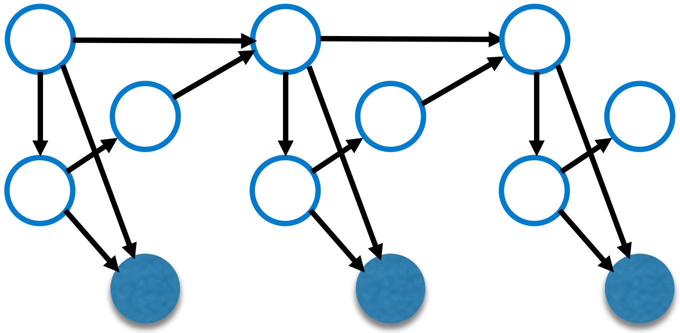
Runtime = **days**

- Adaptive subsampling on **HMM** (*simpler model*)

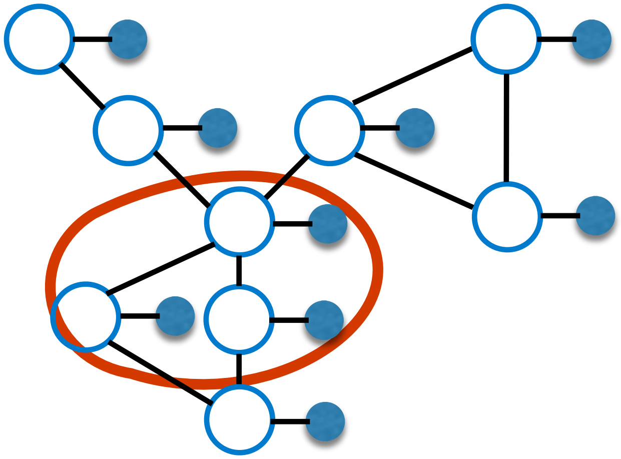
Runtime = under **1 hr**

BNP and Other Extensions

- Presented finite HMM case, but ideas could generalize to:
  - Nonparametric HMMs
  - DBN and MRF models



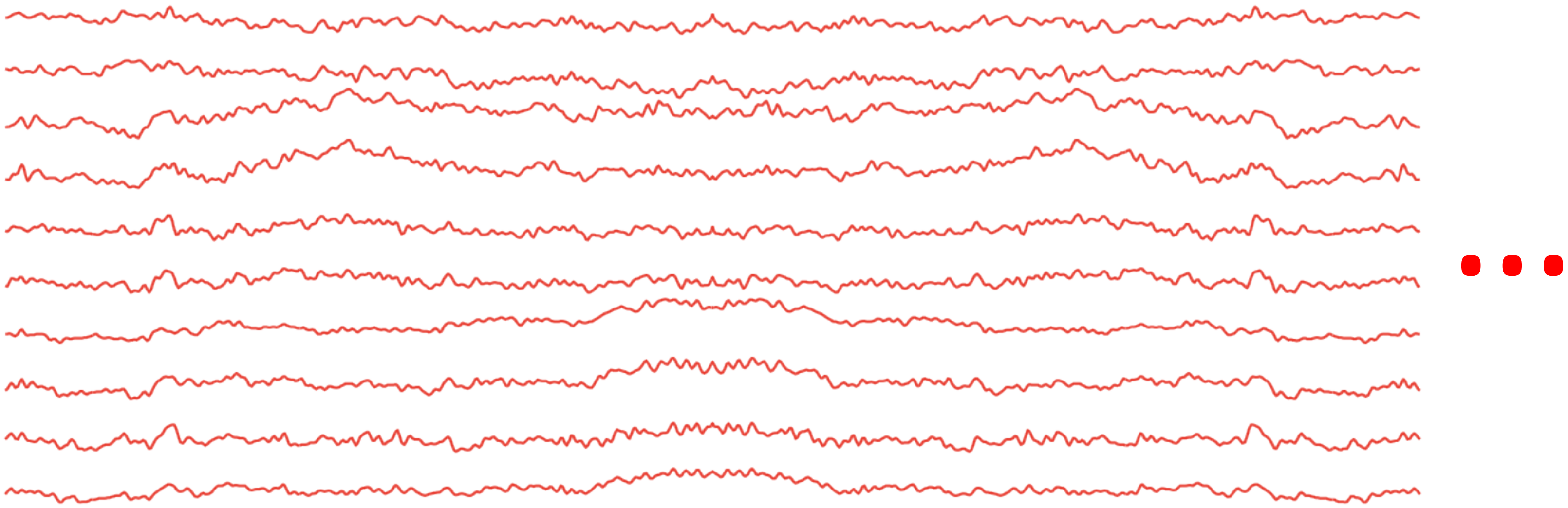
- Applications to:
  - Large spatial fields
  - Spatio-temporal data, etc.



WHAT ABOUT STREAMING DATA?

Issues, solutions, and more issues…

What if data arrive without bound?



Often, not just large dataset, but **streaming**

Assumed Density Filtering

Interested in $p(\theta|x_{1:n})$

- Assume we have $q_{n-1}(\theta) \approx p(\theta|x_{1:n-1})$
- Incorporate new data $\hat{p}(\theta|x_{1:n}) = p(x_n|\theta)q_{n-1}(\theta)$
- Project onto tractable family $\arg\min_{q_n} \mathrm{KL}(\hat{p}||q_n)$

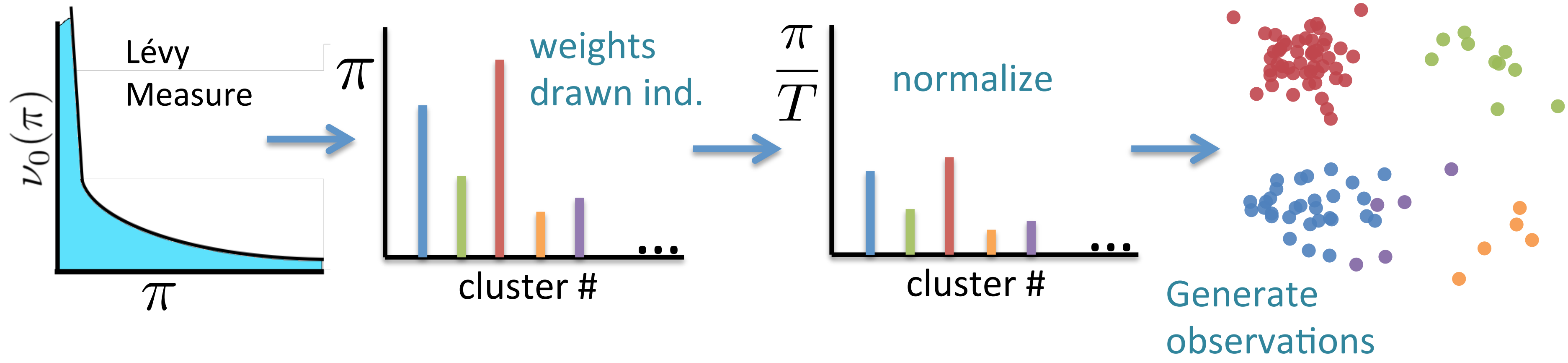
$$\begin{tikzcd}
 & \hat{p}(\theta|x_{1:n-1}) \arrow[d] & & \hat{p}(\theta|x_{1:n}) \arrow[d, "project"] & {} \\
{} \arrow[ur] & q_{n-1}(\theta) \arrow[urr, "predict"] & & q_n(\theta) \arrow[ur] &
\end{tikzcd}$$

Cycling through data multiple times results in the **expectation propagation** algorithm.

Explored ADF for BNP Mixture Models

Bayesian nonparametrics well suited to streaming case since model complexity can adapt

- Existing approaches only for the Dirichlet process (DP)
- We cast DP approach as ADF, and extend to more flexible class of normalized random measures (NRMs)



Tank, Foti, Fox, *AISTATS 2015*

ADF for NRM Mixture Models

Posterior of n data points can be written as a product of factors:

$$p(z_{1:n},\theta|x_{1:n}) \propto p(\theta)\prod_{i=1}^{n} p(x_i|z_i,\theta)p(z_i|z_{1:i-1})$$

likelihood factor: $p(x_i|z_i,\theta)$ predictive factor: $p(z_i|z_{1:i-1})$

Iteratively project onto factorized family $Q_n = \{q; q = \prod_{i=1}^{n} q(z_i)\prod_{k=1}^{\infty} q(\theta_k)\}$

1. Incorporate predictive factor via ADF:

$$\begin{tikzcd}
\boxed{\hat{q}_n(z_{1:n},\theta)} \arrow[r] & \boxed{p(z_{n+1}|z_{1:n})\hat{q}_n(z_{1:n},\theta)} \arrow[r, "Q_{n+1}", "\text{project}"'] & \boxed{q^{\mathrm{pr}}(z_{1:n+1},\theta)} \\
\boxed{p(z_{n+1}|z_{1:n})} \arrow[ur, bend right, "\text{predict}"'] & &
\end{tikzcd}$$

BNP predictive distribution

Only relies on summaries of soft-assignments, rather than full history

$$\boxed{\sum_{i=1}^{t} q_t(z_i = k)}$$

ADF for NRM Mixture Models

Posterior of n data points can be written as a product of factors:

$$p(z_{1:n},\theta|x_{1:n}) \propto p(\theta)\prod_{i=1}^{n} p(x_i|z_i,\theta)p(z_i|z_{1:i-1})$$

likelihood factor: $p(x_i|z_i,\theta)$ predictive factor: $p(z_i|z_{1:i-1})$

Iteratively project onto factorized family $Q_n = \{q; q = \prod_{i=1}^{n} q(z_i) \prod_{k=1}^{\infty} q(\theta_k)\}$

2. Incorporate likelihood via second ADF step:

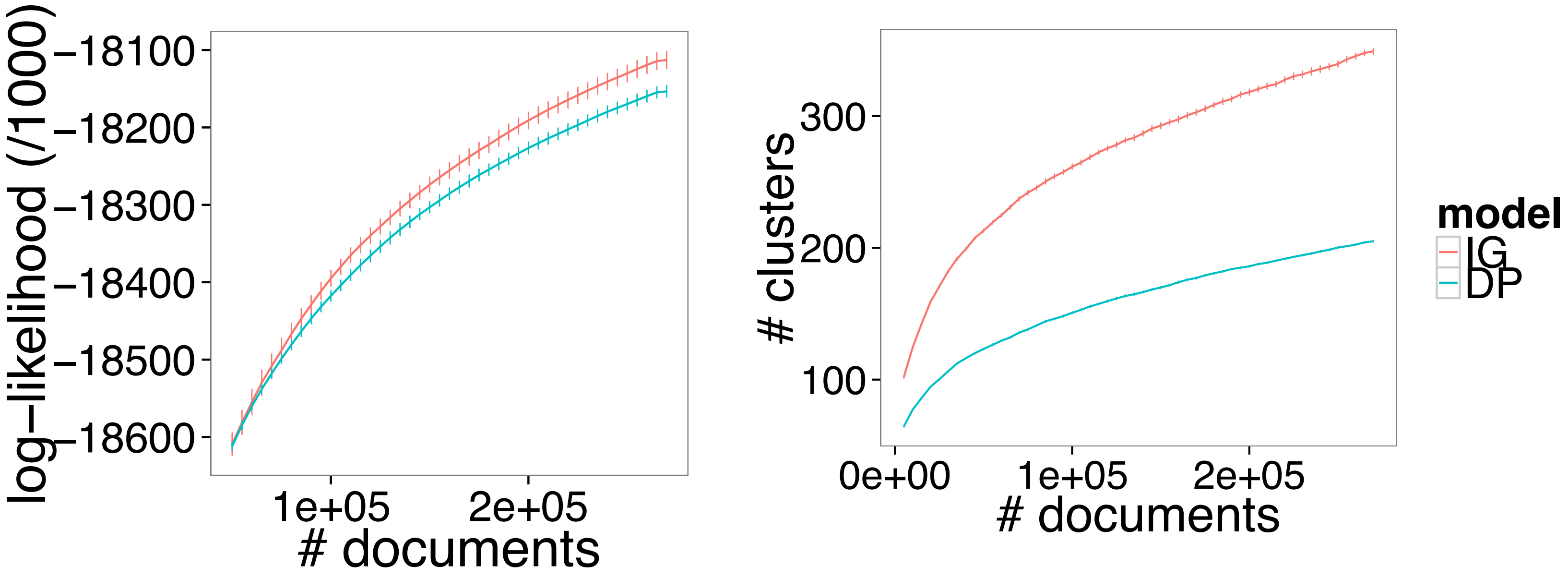
$$\begin{tikzcd}
\boxed{q^{\mathrm{pr}}(z_{1:n+1},\theta)} \arrow[r] & \boxed{p(x_n|z_n,\theta)q^{\mathrm{pr}}(z_{1:n+1},\theta)} \arrow[r, "Q_{n+1}", "\text{project}"'] & \boxed{\hat{q}_{n+1}(z_{1:n+1},\theta)} \\
\boxed{p(x_n|z_n,\theta)} \arrow[ur, bend right, "\text{predict}"'] & &
\end{tikzcd}$$

Typically intractable, so replace with VB update (*reverse KL*)

Similar to what's suggested in Broderick et al. 2013 (SVB)

Online Document Clustering

NYT corpus (N = 266k documents):



Top IG clusters after 1 epoch

Topic 1	Topic 2	Topic 3	Topic 4
athletes (.83)	merger (.36)	reform (.31)	quarterback (.45)
weight (.75)	revenue (3.3)	conservative (.26)	yankees (.45)
exercise (.68)	shares (.31)	senator (.24)	scored (.43)
steroid (.55)	cable (.31)	parties (.22)	pitcher (.38)
supplement (.49)	businesses (.29)	supporter (.22)	offense (.37)

Some challenges…

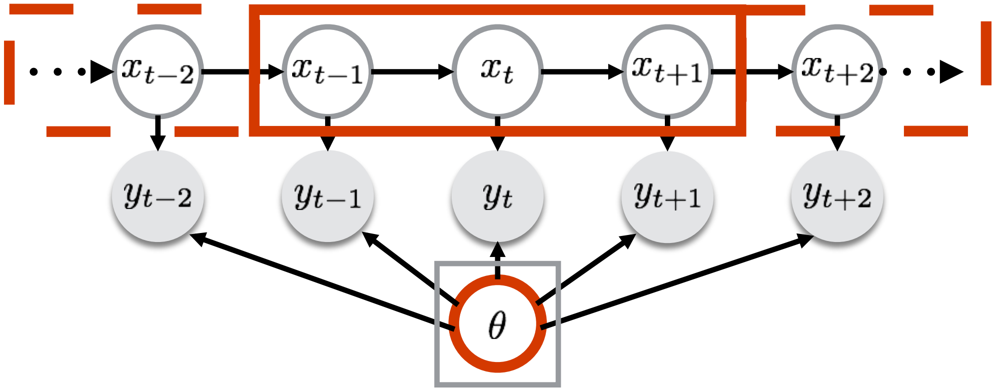
Iterating VB steps leads to more and more concentrated beliefs

- BNP adapts model capacity (i.e., new clusters), which allows one to continue learning
- Don't need to observe all clusters/modes in initial batches
- Harder in HMM case because you have "clusters" and transitions between them…often dwell in one for a while

Theis & Hoffman (2015) trust region approach can help

Summary

- Stochastic variational inference for handling *dependent observations*
  - Harness *memory decay* to form local beliefs on *buffered subchains*
  - Bounding error in approx., can prove *convergence* of iterative algorithm
- Demonstrated on large genomics dataset where batch methods are infeasible



- Discussed promising approaches to streaming case, and challenges for time series data