Stochastic Variational Inference for Hidden Markov Models

Nicholas J. Foti[†], **Jason Xu**[†], **Dillon Laird**, and **Emily B. Fox**
University of Washington
{nfoti@stat,jasonxu@stat,dillonl2@cs,ebfox@stat}.washington.edu

Abstract

Variational inference algorithms have proven successful for Bayesian analysis in large data settings, with recent advances using stochastic variational inference (SVI). However, such methods have largely been studied in independent or exchangeable data settings. We develop an SVI algorithm to learn the parameters of hidden Markov models (HMMs) in a time-dependent data setting. The challenge in applying stochastic optimization in this setting arises from dependencies in the chain, which must be broken to consider minibatches of observations. We propose an algorithm that harnesses the memory decay of the chain to adaptively bound errors arising from edge effects. We demonstrate the effectiveness of our algorithm on synthetic experiments and a large genomics dataset where a batch algorithm is computationally infeasible.

1 Introduction

Modern data analysis has seen an explosion in the size of the datasets available to analyze. Significant progress has been made scaling machine learning algorithms to these massive datasets based on optimization procedures [1, 2, 3]. For example, stochastic gradient descent employs noisy estimates of the gradient based on *minibatches* of data, avoiding a costly gradient computation using the full dataset [4]. There is considerable interest in leveraging these methods for Bayesian inference since traditional algorithms such as Markov chain Monte Carlo (MCMC) scale poorly to large datasets, though subset-based MCMC methods have been recently proposed as well [5, 6, 7, 8].

Variational Bayes (VB) casts posterior inference as a tractable optimization problem by minimizing the Kullback-Leibler divergence between the target posterior and a family of simpler *variational distributions*. Thus, VB provides a natural framework to incorporate ideas from stochastic optimization to perform scalable Bayesian inference. Indeed, a scalable modification to VB harnessing stochastic gradients—*stochastic variational inference* (SVI)—has recently been applied to a variety of Bayesian latent variable models [9, 10]. Minibatch-based VB methods have also proven effective in a streaming setting where data arrives sequentially [11].

However, these algorithms have been developed assuming independent or exchangeable data. One exception is the SVI algorithm for the mixed-membership stochastic block model [12], but independence at the level of the generative model must be exploited. SVI for Bayesian time series including HMMs was recently considered in settings where each minibatch is a set of *independent* series [13], though in this setting again dependencies do not need to be broken.

In contrast, we are interested in applying SVI to very long time series. As a motivating example, consider the application in Sec. 4 of a genomics dataset consisting of $T = 250$ million observations in 12 dimensions modeled via an HMM to learn human chromatin structure. An analysis of the entire sequence is computationally prohibitive using standard Bayesian inference techniques for

[†] Co-first authors contributed equally to this work.

HMMs due to a per-iteration complexity linear in T. Unfortunately, despite the simple chain-based dependence structure, applying a minibatch-based method is not obvious. In particular, there are two potential issues immediately arising in sampling subchains as minibatches: (1) the subsequences are not mutually independent, and (2) updating the latent variables in the subchain ignores the data outside of the subchain introducing error. We show that for (1), appropriately scaling the noisy subchain gradients preserves unbiased gradient estimates. To address (2), we propose an approximate message-passing scheme that adaptively bounds error by accounting for memory decay of the chain.

We prove that our proposed *SVIHMM* algorithm converges to a local mode of the batch objective, and empirically demonstrate similar performance to batch VB in significantly less time on synthetic datasets. We then consider our genomics application and show that SVIHMM allows efficient Bayesian inference on this massive dataset where batch inference is computationally infeasible.

2 Background

2.1 Hidden Markov models

Hidden Markov models (HMMs) [14] are a class of discrete-time doubly stochastic processes consisting of observations y_t and latent states $x_t \in \{1, \dots, K\}$ generated by a discrete-valued Markov chain. Specifically, for $\mathbf{y} = (y_1, \dots, y_T)$ and $\mathbf{x} = (x_1, \dots, x_T)$, the joint distribution factorizes as

$$p(\mathbf{x}, \mathbf{y}) = \pi_0(x_1)p(y_1|x_1)\prod_{t=2}^{T} p(x_t|x_{t-1}, A)p(y_t|x_t, \phi) \tag{1}$$

where $A = [A_{ij}]_{i,j=1}^K$ is the *transition matrix* with $A_{ij} = \Pr(x_t = j|x_{t-1} = i)$, $\phi = \{\phi_k\}_{k=1}^K$ the *emission parameters*, and π_0 the *initial distribution*. We denote the set of HMM parameters as $\theta = (\pi_0, A, \phi)$. We assume that the underlying chain is irreducible and aperiodic so that a *stationary distribution* π exists and is unique. Furthermore, we assume that we observe the sequence at stationarity so that $\pi_0 = \pi$, where π is given by the leading left-eigenvector of A. As such, we do not seek to learn π_0 in the setting of observing a single realization of a long chain.

We specify conjugate Dirichlet priors on the rows of the transition matrix as

$$p(A) = \prod_{j=1}^{K} \text{Dir}(A_{i:} \mid \alpha_j^A). \tag{2}$$

Here, $\text{Dir}(\pi \mid \alpha)$ denotes a K-dimensional Dirichlet distribution with concentration parameters α. Although our methods are more broadly applicable, we focus on HMMs with multivariate Gaussian emissions where $\phi_k = \{\mu_k, \Sigma_k\}$, with conjugate normal-inverse-Wishart (NIW) prior

$$y_t \mid x_t \sim N(y_t \mid \mu_{x_t}, \Sigma_{x_t}), \qquad \phi_k = (\mu_k, \Sigma_k) \sim \text{NIW}(\mu_0, \kappa_0, \Sigma_0, \nu_0). \tag{3}$$

For simplicity, we suppress dependence on θ and write $\pi(x_0)$, $p(x_t|x_{t-1})$, and $p(y_t|x_t)$ throughout.

2.2 Structured mean-field VB for HMMs

We are interested in the *posterior distribution* of the state sequence and parameters given an observation sequence, denoted $p(\mathbf{x}, \theta|\mathbf{y})$. While evaluating marginal likelihoods, $p(\mathbf{y}|\theta)$, and most probable state sequences, $\arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}, \theta)$, are tractable via the forward-backward (FB) algorithm when parameter values θ are *fixed* [14], exact computation of the posterior is intractable for HMMs. Markov chain Monte Carlo (MCMC) provides a widely used sampling-based approach to posterior inference in HMMs [15, 16]. We instead focus on variational Bayes (VB), an optimization-based approach that approximates $p(\mathbf{x}, \theta|\mathbf{y})$ by a variational distribution $q(\theta, \mathbf{x})$ within a simpler family. Typically, for HMMs a *structured mean field* approximation is considered:

$$q(\theta, \mathbf{x}) = q(A)q(\phi)q(\mathbf{x}), \tag{4}$$

breaking dependencies only between the parameters $\theta = \{A, \phi\}$ and latent state sequence $\mathbf{x}$ [17]. Note that making a full mean field assumption in which $q(\mathbf{x}) = \prod_{i=1}^{T} q(x_i)$ loses crucial information about the latent chain needed for accurate inference.

Each factor in Eq. (4) is endowed with its own variational parameter and is set to be in the same exponential family distribution as its respective complete conditional. The variational parameters are optimized to maximize the *evidence lower bound* (ELBO) $\mathcal{L}$:

$$\ln p(\mathbf{y}) \geq E_q\left[\ln p(\theta)\right] - E_q\left[\ln q(\theta)\right] + E_q\left[\ln p(\mathbf{y}, \mathbf{x}|\theta)\right] - E_q\left[\ln q(\mathbf{x})\right] := \mathcal{L}(q(\theta), q(\mathbf{x})). \quad (5)$$

Maximizing $\mathcal{L}$ is equivalent to minimizing the KL divergence $\mathrm{KL}(q(\mathbf{x}, \theta)||p(\mathbf{x}, \theta|\mathbf{y}))$ [18]. In practice, we alternate updating the *global parameters* θ—those coupled to the entire set of observations—and the *local variables* $\{x_t\}$—a variable corresponding to each observation, y_t. Details on computing the terms in the equations and algorithms that follow are in the Supplement.

The *global update* is derived by differentiating $\mathcal{L}$ with respect to the global variational parameters [17]. Assuming a conjugate exponential family leads to a simple coordinate ascent update [9]:

$$\mathbf{w} = \mathbf{u} + E_{q(\mathbf{x})}\left[t(\mathbf{x}, \mathbf{y})\right]. \quad (6)$$

Here, $t(\mathbf{x}, \mathbf{y})$ denotes the vector of sufficient statistics, and $\mathbf{w} = (\mathbf{w}^A, \mathbf{w}^\phi)$ and $\mathbf{u} = (\mathbf{u}^A, \mathbf{u}^\phi)$ the variational parameters and model hyperparameters, respectively, in natural parameter form.

The *local update* is derived analogously, yielding the optimal variational distribution over the latent sequence:

$$q^*(\mathbf{x}) \propto \exp\left(E_{q(A)}\left[\ln \pi(x_1)\right] + \sum_{t=2}^{T} E_{q(A)}\left[\ln A_{x_{t-1}, x_t}\right] + \sum_{t=1}^{T} E_{q(\phi)}\left[\ln p(y_t|x_t)\right]\right). \quad (7)$$

Compare with Eq. (1). Here, we have replaced probabilities by exponentiated expected log probabilities under the current variational distribution. To determine the optimal $q^*(\mathbf{x})$ in Eq. (7), define:

$$\widetilde{A}_{j,k} := \exp\left[E_{q(A)} \ln(A_{j,k})\right] \quad \widetilde{p}(y_t|x_t = k) := \exp\left[E_{q(\phi)} \ln p(y_t|x_t = k)\right]. \quad (8)$$

We estimate π with $\hat{\pi}$ being the leading eigenvector of $E_{q(A)}[A]$. We then use $\hat{\pi}, \tilde{A} = (\widetilde{A}_{j,k})$, and $\tilde{p} = \{\widetilde{p}(y_t|x_t = k), k = 1, \ldots, K, t = 1, \ldots, T\}$ to run a forward-backward algorithm, producing forward messages α and backward messages β which allow us to compute $q^*(x_t = k)$ and $q^*(x_{t-1} = j, x_t = k)$. [19, 17]. See the Supplement.

2.3 Stochastic variational inference for non-sequential models

Even in non-sequential models, the batch VB algorithm requires an entire pass through the dataset for each update of the global parameters. This can be costly in large datasets, and wasteful when local-variable passes are based on uninformed initializations of the global parameters or when many data points contain redundant information.

To cope with this computational challenge, *stochastic variational inference* (SVI) [9] leverages a Robbins-Monro algorithm [1] to optimize the ELBO via stochastic gradient ascent. When the data are independent, the ELBO in Eq. (5) can be expressed as

$$\mathcal{L} = E_{q(\theta)}\left[\ln p(\theta)\right] - E_{q(\theta)}\left[\ln q(\theta)\right] + \sum_{i=1}^{T} E_{q(x_i)}\left[\ln p(y_i, x_i|\theta)\right] - E_{q(\mathbf{x})}\left[\ln q(\mathbf{x})\right]. \quad (9)$$

If a single observation index s is sampled uniformly $s \sim \mathrm{Unif}(1, \ldots, T)$, the ELBO corresponding to (x_s, y_s) as if it were replicated T times is given by

$$\mathcal{L}^s = E_{q(\theta)}\left[\ln p(\theta)\right] - E_{q(\theta)}\left[\ln q(\theta)\right] + T \cdot \left(E_{q(x_s)}\left[\ln p(y_s, x_s|\theta)\right] - E_{q(x_s)}\left[\ln q(x_s)\right]\right), \quad (10)$$

and it is clear that $E_s[\mathcal{L}^s] = \mathcal{L}$. At each iteration n of the SVI algorithm, a data point y_s is sampled and its *local* $q^*(x_s)$ is computed given the current estimate of global variational parameters $\mathbf{w}_n$. Next, the *global update* is performed via a noisy, unbiased gradient step ($E_s[\hat{\nabla}_{\mathbf{w}} \mathcal{L}^s] = \nabla_{\mathbf{w}} \mathcal{L}$). When all pairs of distributions in the model are conditionally conjugate, it is cheaper to compute the stochastic *natural gradient*, $\widetilde{\nabla}_{\mathbf{w}} \mathcal{L}^s$, which additionally accounts for the information geometry of the distribution [9]. The resulting stochastic natural gradient step with step-size ρ_n is:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \rho_n \widetilde{\nabla}_{\mathbf{w}} \mathcal{L}^s(\mathbf{w}_n). \quad (11)$$

We show the form of $\widetilde{\nabla}_{\mathbf{w}} \mathcal{L}^s$ in Sec. 3.2, specifically in Eq. (13) with details in the Supplement.

3 Stochastic variational inference for HMMs

The batch VB algorithm of Sec. 2.2 becomes prohibitively expensive as the length of the chain T becomes large. In particular, the forward-backward algorithm in the local step takes $O(K^2T)$ time. Instead, we turn to a subsampling approach, but naively applying SVI from Sec. 2.3 fails in the HMM setting: decomposing the sum over local variables into a sum of independent terms as in Eq. (9) ignores crucial transition counts, equivalent to making a full mean-field approximation.

Extending SVI to HMMs requires additional considerations due to the dependencies between the observations. It is clear that *subchains* of consecutive observations rather than individual observations are necessary to capture the transition structure (see Sec. 3.1). We show that if the local variables of each subchain can be exactly optimized, then stochastic gradients computed on subchains can be scaled to preserve unbiased estimates of the full gradient (see Sec. 3.2).

Unfortunately, as we show in Sec. 3.3, the local step becomes approximate due to edge effects: local variables are incognizant of nodes outside of the subchain during the forward-backward pass. Although an *exact* scheme requires message passing along the entire chain, we harness the memory decay of the latent Markov chain to guarantee that local state beliefs in each subchain form an ϵ-*approximation* $q_\epsilon(\mathbf{x})$ to the full-data beliefs $q^*(\mathbf{x})$. We achieve these approximations by adaptively buffering the subchains with extra observations based on current global parameter estimates. We then prove that for ϵ sufficiently small, the noisy gradient computed using $q_\epsilon(\mathbf{x})$ corresponds to an ascent direction in $\mathcal{L}$, guaranteeing convergence of our algorithm to a local optimum. We refer to our algorithm, which is outlined in Alg. 1, as *SVIHMM*.

Algorithm 1 Stochastic Variational Inference for HMMs (SVIHMM)

1: Initialize variational parameters $(\mathbf{w}_0^A, \mathbf{w}_0^\phi)$ and choose stepsize schedule ρ_n, $n = 1, 2, \ldots$
2: **while** (convergence criterion is not met) **do**
3: Sample a subchain $\mathbf{y}^S \subset \{y_1, \ldots, y_T\}$ with $S \sim p(S)$
4: **Local step:** Compute $\hat{\pi}, \widetilde{A}, \widetilde{p}_S$ and run $q(\mathbf{x}^S) = \texttt{ForwardBackward}(\mathbf{y}^S, \hat{\pi}, \widetilde{A}, \widetilde{p}_S)$.
5: **Global update:** $\mathbf{w}_{n+1} = \mathbf{w_n}(1-\rho_n) + \rho_n(\mathbf{u} + \mathbf{c}^T E_{q(\mathbf{x}^S)}[t(\mathbf{x}^S, \mathbf{y}^S)])$
6: **end while**

3.1 ELBO for subsets of data

Unlike the independent data case (Eq. (9)), the local term in the HMM setting decomposes as

$$\ln p(\mathbf{y}, \mathbf{x}|\theta) = \ln \pi(x_1) + \sum_{t=2}^{T} \ln A_{x_{t-1}, x_t} + \sum_{i=1}^{T} \ln p(y_t|x_t). \tag{12}$$

Because of the paired terms in the first sum, it is necessary to consider *consecutive observations* to learn transition structure. For the SVIHMM algorithm, we define our basic sampling unit as *subchains* $\mathbf{y}^S = (y_1^S, \ldots, y_L^S)$, where S refers to the associated indices. We denote the ELBO restricted to $\mathbf{y}^S$ as $\mathcal{L}^S$, and associated natural gradient as $\widetilde{\nabla}_{\mathbf{w}} \mathcal{L}^S$.

3.2 Global update

We detail the global update assuming we have optimized $q^*(\mathbf{x})$ *exactly* (i.e., as in the batch setting), although this assumption will be relaxed as discussed in Sec 3.3. Paralleling Sec. 2.3, the global SVIHMM step involves updating the global variational parameters $\mathbf{w}$ via stochastic (natural) gradient ascent based on $q^*(\mathbf{x}^S)$, the beliefs corresponding to our current subchain S.

Recall from Eq. (10) that the original SVI algorithm maintains $E_s[\widetilde{\nabla}_{\mathbf{w}} \mathcal{L}^s] = \widetilde{\nabla}_{\mathbf{w}} \mathcal{L}$ by scaling the gradient based on an individual observation s by the total number of observations T. In the HMM case, we analogously derive a *batch factor* vector $\mathbf{c} = (c^A, c^\phi)$ such that

$$E_S[\widetilde{\nabla}_{\mathbf{w}} \mathcal{L}^S] = \widetilde{\nabla}_{\mathbf{w}} \mathcal{L} \quad \text{with} \quad \widetilde{\nabla}_{\mathbf{w}} \mathcal{L}^S = \mathbf{u} + \mathbf{c}^T E_{q^*(\mathbf{x}^S)} \left[t(\mathbf{x}^S, \mathbf{y}^S) \right] - \mathbf{w}. \tag{13}$$

The specific form of Eq. (13) for Gaussian emissions is in the Supplement. Now, the Robbins-Monro average in Eq. (11) can be written as

$$\mathbf{w}_{n+1} = \mathbf{w}_n(1-\rho_n) + \rho_n(\mathbf{u} + \mathbf{c}^T E_{q^*(\mathbf{x}^S)}[t(\mathbf{x}^S, \mathbf{y}^S)]). \tag{14}$$

When the noisy natural gradients $\widetilde{\nabla}_{\mathbf{w}}\mathcal{L}^S$ are independent and unbiased estimates of the true natural gradient, the iterates in Eq. (14) converge to a local maximum of $\mathcal{L}$ under mild regularity conditions as long as step-sizes ρ_n satisfy $\sum_n \rho_n^2 < \infty$, and $\sum_n \rho_n = \infty$ [2, 9]. In our case, the noisy gradients are necessarily correlated even for independently sampled subchains due to dependence between observations $(y_1, \ldots, y_T)$. However, as detailed in [20], unbiasedness suffices for convergence of Eq. (14) to a local mode.

Batch factor Recalling our assumption of being at stationarity, $E_{q(\pi)} \ln \pi(x_1) = E_{q(\pi)} \ln \pi(x_i)$ for all i. If we sample subchains from the uniform distribution over subchains of length L, denoted $p(S)$, then we can write

$$E_S\left[E_q \ln p(\mathbf{y}^S, \mathbf{x}^S|\theta)\right] \approx p(S) E_q \left[\sum_{t=1}^{T-L+1} \ln \pi(x_t) + (L-1)\sum_{t=2}^{T} \ln A_{x_{t-1},x_t} + L\sum_{t=1}^{T} p(y_t|x_t)\right], \tag{15}$$

where the expectation is with respect to (π, A, ϕ); this is detailed in the Supplement. The approximate equality in Eq. (15) arises because while most transitions appear in $L-1$ subchains, those near the endpoints of the full chain do not, e.g., x_1 and x_T appear in only one subchain. This error becomes negligible as the length of the HMM increases. Since $p(S)$ is uniform over all length L subchains, by linearity of expectation the batch factor $\mathbf{c} = (c^A, c^\phi)$ is given by $c^A = (T-L+1)/(L-1)$, $c^\phi = (T-L+1)/L$. Other choices of $p(S)$ can be used by considering the appropriate version of Eq. (15) analogously to [12], generally with a batch factor $\mathbf{c}^S$ varying with each subset $\mathbf{y}^S$.

3.3 Local update

The optimal SVIHMM local variational distribution arises just as in the batch case of Eq. (7), but with time indices restricted to the length L subchain $\mathbf{y}^S$:

$$q^*(\mathbf{x}^S) \propto \exp\left(E_{q(A)}\left[\ln \pi(x_1^S)\right] + \sum_{\ell=2}^{L} E_{q(A)}\left[\ln A_{x_{\ell-1}^S, x_\ell^S}\right] + \sum_{\ell=1}^{L} E_{q(\phi)}\left[\ln p(y_\ell^S|x_\ell^S)\right]\right). \tag{16}$$

To compute these local beliefs, we use our current $q(A), q(\phi)$—which have been informed by all previous subchains—to form $\hat{\pi}$, $\widetilde{A}$, $\widetilde{p}_S = \{\widetilde{p}(y_\ell^S|x_\ell^S = k), \forall k, \ell = 1, \ldots, L\}$, with these parameters defined as in the batch case. We then use these parameters in a forward-backward algorithm detailed in the Supplement. However, this message passing produces only an approximate optimization due to loss of information incurred at the ends of the subchain. Specifically, for $\mathbf{y}^S = (y_t, \ldots, y_{t+L})$, the forward messages coming from $y_1, \ldots, y_{t-1}$ are not available to y_t, and similarly the backwards messages from $y_{t+L+1}, \ldots, y_T$ are not available to y_{t+L}.

Recall our assumption in the global update step that $q^*(\mathbf{x}^S)$ corresponds to a subchain of the full-data optimal beliefs $q^*(\mathbf{x})$. Here, we see that this assumption is assuredly false; instead, we analyze the implications of using approximate local subchain beliefs and aim to ameliorate the edge effects.

Buffering subchains To cope with the subchain edge effects, we augment the subchain S with enough extra observations on each end so that the local state beliefs, $q(x_i)$, $i \in S$, are within an ϵ-ball of $q^*(x_i)$ — those had we considered the entire chain. The practicality of this approach arises from the approximate finite memory of the process. In particular, consider performing a forward-backward pass on $(x_{1-\tau}^S, \ldots, x_{L+\tau}^S)$ leading to approximate beliefs $\tilde{q}^\tau(x_i)$. Given $\epsilon > 0$, define τ_ϵ as the smallest buffer length τ such that

$$\max_{i \in S} ||\tilde{q}^\tau(x_i) - q^*(x_i)||_1 \leq \epsilon. \tag{17}$$

The τ that satisfies Eq. (17) determines the number of observations used to *buffer* the subchain. After improving subchain beliefs, we discard $\tilde{q}^\tau(x_i)$, $i \in$ *buffer*, prior to the global update. As will be seen in Sec. 4, in practice the necessary τ_ϵ is typically very small relative to the lengthy observation sequences of interest.

Buffering subchains is related to *splash belief propagation* (BP) for parallel inference in undirected graphical models, where the belief at any given node is monitored based on locally-aware message passing in order to maintain a good approximation to the true belief [21]. Unlike splash BP, we

embed the buffering scheme inside an iterative procedure for updating both the local latent structure and the global parameters, which affects the ϵ-approximation in future iterations. Likewise, we wish to maintain the approximation on an entire subchain, not just at a single node.

Even in settings where parameters θ are known, as in splash BP, analytically choosing τ_ϵ is generally infeasible. As such, we follow the approach of splash BP to select an approximate τ_ϵ. We then go further by showing that SVIHMM still converges using approximate messages within an uncertain parameter setting where θ is learned simultaneously with the state sequence $\mathbf{x}$.

Specifically, we approximate τ_ϵ by monitoring the change in belief residuals with a sub-routine `GrowBuf`, outlined in Alg. 2, that iteratively expands a buffer $q^{\text{old}} \rightarrow q^{\text{new}}$ around a given subchain $\mathbf{y}^S$. `Growbuf` terminates when all belief residuals satisfy

$$\max_{i \in S} ||q(x_i)^{\text{new}} - q(x_i)^{\text{old}}||_1 \leq \epsilon. \tag{18}$$

The `GrowBuf` sub-routine can be computed efficiently due to (1) monotonicity of the forward and backward messages so that only residuals at endpoints, $q(x_1^S)$ and $q(x_L^S)$, need be considered, and (2) the reuse of computations. Specifically, the forward-backward pass can be rooted at the midpoint of $\mathbf{y}^S$ so that messages to the endpoints can be efficiently propagated, and vice versa [22].

Furthermore, choosing sufficiently small ϵ guarantees that the noisy natural gradient lies in the same half-plane as the true natural gradient, a sufficient condition for maintaining convergence when using approximate gradients [23]; the proof is presented in the Supplement.

Algorithm 2 `GrowBuf` procedure.

Input: subchain S, min buffer length $u \in \mathbb{Z}_+$, error tolerance $\epsilon > 0$.
Initialize $q^{\text{old}}(\mathbf{x}^S) = \texttt{ForwardBackward}(\mathbf{y}^S, \hat{\pi}, \widetilde{A}, \widetilde{p}_S)$ and set $S^{\text{old}} = S$.
while true **do**
  Grow buffer S^{new} by extending S^{old} by u observations in each direction.
  $q^{\text{new}}(\mathbf{x}^{S^{\text{new}}}) = \texttt{ForwardBackward}(\mathbf{y}^{S^{\text{new}}}, \hat{\pi}, \widetilde{A}, \widetilde{p}_{S^{\text{new}}})$, reusing messages from S^{old}.
  if $\left|\left|q^{\text{new}}(\mathbf{x}^S) - q^{\text{old}}(\mathbf{x}^S)\right|\right| < \epsilon$ **then**
    return $q^*(\mathbf{x}^S) = q^{\text{new}}(\mathbf{x}^S)$
  end if
  Set $S^{\text{old}} = S^{\text{new}}$ and $q^{\text{old}} = q^{\text{new}}$.
end while

3.4 Minibatches for variance mitigation and their effect on computational complexity

Stochastic gradient algorithms often benefit from sampling multiple observations in order to reduce the variance of the gradient estimates at each iteration. We use a similar idea in SVIHMM by sampling a *minibatch* $B = (\mathbf{y}^{S_1}, \ldots, \mathbf{y}^{S_M})$ consisting of M subchains. If the latent Markov chain tends to dwell in one component for extended periods, sampling one subchain may only contain information about a select number of states observed in that component. Increasing the length of this subchain may only lead to redundant information from this component. In contrast, using a minibatch of many smaller subchains may discover disparate components of the chain at comparable computational cost, accelerating learning and leading to a better local optimum. However, subchains must be sufficiently long to be informative of transition dynamics. In this setting, the local step on each subchain is identical; summing over subchains in the minibatch yields the gradient update:

$$\hat{\mathbf{w}}^B = \sum_{S \in B} \mathbf{c}^T E_{q(\mathbf{x}^S)} \left[t(\mathbf{x}^S, \mathbf{y}^S) \right], \quad \mathbf{w}_{n+1} = \mathbf{w}_n(1 - \rho_n) + \rho_n \left(u + \frac{\hat{\mathbf{w}}^B}{|B|} \right).$$

We see that the computational complexity of SVIHMM is $O(K^2(L + 2\tau_\epsilon)M)$, leading to significant efficiency gains compared to $O(K^2 T)$ in batch inference when $(L + 2\tau_\epsilon)M << T$.

4 Experiments

We evaluate the performance of SVIHMM compared to batch VB on synthetic experiments designed to illustrate the trade off between the choice of subchain length L and the number of subchains per

Table 1: Runtime and predictive log-probability (without `GrowBuf`) on RC data.

$\lfloor L/2 \rfloor$	Runtime (sec.)	Avg. iter. time (sec.)	log-predictive
100	2.74 ± 0.001	0.03 ± 0.000	-5.915 ± 0.004
500	11.79 ± 0.004	0.12 ± 0.000	-5.850 ± 0.000
1000	23.17 ± 0.006	0.23 ± 0.000	-5.850 ± 0.000
batch	1240.73 ± 0.370	248.15 ± 0.074	-5.840 ± 0.000

minibatch M. We also demonstrate the utility of `GrowBuf`. We then apply our algorithm to gene segmentation in a large human chromatin data set.

Synthetic data We create two synthetic datasets with $T = 10,000$ observations and $K = 8$ latent states. The first, called *diagonally dominant* (DD), illustrates the potential benefit of large M, the number of sampled subchains per minibatch. The Markov chain heavily self-transitions so that most subchains contain redundant information with observations generated from the same latent state. Although transitions are rarely observed, the emission means are set to be distinct so that this example is likelihood-dominated and highly identifiable. Thus, fixing a computational budget, we expect large M to be preferable to large L, covering more of the observation sequence and avoiding poor local modes arising from redundant information.

The second dataset we consider contains two *reversed cycles* (RC): the Markov chain strongly transitions from states $1 \to 2 \to 3 \to 1$ and $5 \to 7 \to 6 \to 5$ with a small probability of transitioning between cycles via bridge states 4 and 8. The emission means for the two cycles are very similar but occur in reverse order with respect to the transitions. Transition information in observing long enough dynamics is thus crucial to identify between states $1, 2, 3$ and $5, 6, 7$, and a large enough L is imperative. The Supplement contains details for generating both synthetic datasets.

We compare SVIHMM to batch VB on these two synthetic examples. For each per parameter setting, we ran 20 random restarts of SVIHMM for 100 iterations and batch VB until convergence of the ELBO. A *forgetting rate* κ parametrizes step sizes $\rho_n = (1+n)^{-\kappa}$. We fix the total number of observations $L \times M$ used per iteration of SVIHMM such that increasing M implies decreasing L (and vice versa).

In Fig. 1(a) we compare $||\hat{A} - A||_F$, where A is the true transition matrix and $\hat{A}$ its learned variational mean. We see trends one would expect: the small L, large M settings achieve better performance for the DD example, but the opposite holds for RC, with $\lfloor L/2 \rfloor = 1$ significantly underperforming. (Of course, allowing large L *and* M is always preferable, except computationally.) Under appropriate settings in both cases, we achieve comparable performance to batch VB. In Fig. 1(b), we see similar trends in terms of predictive log-probability holding out 10% of the observations as a test set and using 5-fold cross validation. Here, we actually notice that SVIHMM often achieves *higher* predictive log-probability than batch VB, which is attributed to the fact that stochastic algorithms can find better local modes than their non-random counterparts.

A timing comparison of SVIHMM to batch VB with $T = 3$ million is presented in Table 4. All settings of SVIHMM run faster than even a single iteration of batch, with only a negligible change in predictive log-likelihood. Further discussion on these timing results is in the Supplement.

Motivated by the demonstrated importance of choice of L, we now turn to examine the impact of the `GrowBuf` routine via predictive log-probability. In Fig. 1(b), we see a noticeable improvement for small L settings when `GrowBuf` is incorporated (the dashed lines in Fig. 1(b)). In particular, the RC example is now learning dynamics of the chain even with $\lfloor L/2 \rfloor = 1$, which was not possible without buffering. `GrowBuf` thus provides robustness by guarding against poor choice of L. We note that the buffer routine does not overextend subchains, on average growing by only ≈ 8 observations with $\epsilon = 1 \times 10^{-6}$. Since the number of observations added is usually small, `GrowBuf` does not significantly add to per-iteration computational cost (see the Supplement).

Human chromatin segmentation We apply the SVIHMM algorithm to a massive human chromatin dataset provided by the ENCODE project [24]. This data was studied in [25] with the goal of unsupervised pattern discovery via *segmentation* of the genome. Regions sharing the same labels have certain common properties in the observed data, and because the labeling at each position is unknown but influenced by the label at the previous position, an HMM is a natural model [26].

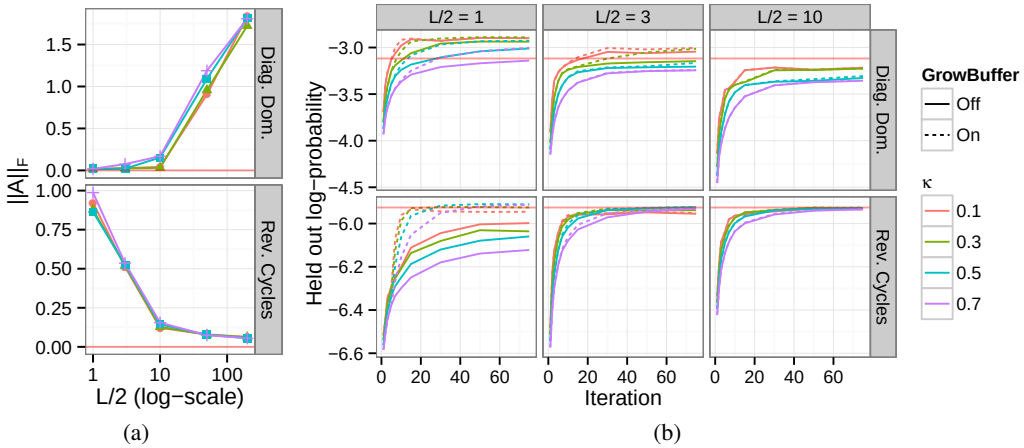

Figure 1: (a) Transition matrix error varying L with $L \times M$ fixed. (b) Effect of incorporating `GrowBuf`. Batch results denoted by horizontal red line in both figures.

We were provided with 250 million observations consisting of twelve assays carried out in the chronic myeloid leukemia cell line K562. We analyzed the data using SVIHMM on an HMM with 25 states and 12 dimensional Gaussian emissions. We compare our performance to the corresponding segmentation learned by an expectation maximization (EM) algorithm applied to a more flexible dynamic Bayesian network model (DBN) [27]. Due to the size of the dataset, the analysis of [27] requires breaking the chain into several blocks, severing long range dependencies.

We assess performance by comparing the false discovery rate (FDR) of predicting active promoter elements in the sequence. The lowest (best) FDR achieved with SVIHMM over 20 random restarts trials was .999026 using $\lfloor L/2 \rfloor = 2000, M = 50, \kappa = .5$[1], comparable and slightly lower than the .999038 FDR obtained using DBN-EM on the severed data [27]. We emphasize that even when restricted to a simpler HMM model, learning on the full data via SVIHMM attains similar results to that of [27] with significant gains in efficiency. In particular, our SVIHMM runs require only under an hour for a fixed 100 iterations, the maximum iteration limit specified in the DBN-EM approach. In contrast, even with a parallelized implementation over the broken chain, the DBN-EM algorithm can take days. In conclusion, SVIHMM enables scaling to the entire dataset, allowing for a more principled approach by utilizing the data jointly.

5 Discussion

We have presented stochastic variational inference for HMMs, extending such algorithms from independent data settings to handle time dependence. We elucidated the complications that arise when sub-sampling dependent observations and proposed a scheme to mitigate the error introduced from breaking dependencies. Our approach provides an adaptive technique with provable guarantees for convergence to a local mode. Further extensions of the algorithm in the HMM setting include adaptively selecting the length of meta-observations and parallelizing the local step when the number of meta-observations is large. Importantly, these ideas generalize to other settings and can be applied to Bayesian nonparametric time series models, general state space models, and other graph structures with spatial dependencies.

Acknowledgements

This work was supported in part by the TerraSwarm Research Center sponsored by MARCO and DARPA, DARPA Grant FA9550-12-1-0406 negotiated by AFOSR, and NSF CAREER Award IIS-1350133. JX was supported by an NDSEG fellowship. We also appreciate the data, discussions, and guidance on the ENCODE project provided by Max Libbrecht and William Noble.

[1] Other parameter settings were explored.

References

[1] H. Robbins and S. Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[2] L. Bottou. Online algorithms and stochastic approximations. In *Online Learning and Neural Networks*. Cambridge University Press, 1998.

[3] L. Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. In *International Conference on Computational Statistics*, pages 177–187, August 2010.

[4] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. on Optimization*, 19(4):1574–1609, January 2009.

[5] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, pages 681–688, 2011.

[6] D. Maclaurin and R. P. Adams. Firefly Monte Carlo: Exact MCMC with subsets of data. *CoRR*, abs/1403.5693, 2014.

[7] X. Wang and D. B. Dunson. Parallelizing MCMC via Weierstrass sampler. *CoRR*, abs/1312.4605, 2014.

[8] W. Neiswanger, C. Wang, and E. Xing. Asymptotically exact, embarrassingly parllel MCMC. *CoRR*, abs/1311.4780, 2014.

[9] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, May 2013.

[10] M. Bryant and E. B. Sudderth. Truly nonparametric online variational inference for hierarchical Dirichlet processes. In *Advances in Neural Information Processing Systems*, pages 2708–2716, 2012.

[11] T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, pages 1727–1735, 2013.

[12] P. Gopalan, D. M. Mimno, S. Gerrish, M. J. Freedman, and D. M. Blei. Scalable inference of overlapping communities. In *Advances in Neural Information Processing Systems*, pages 2258–2266, 2012.

[13] M. J. Johnson and A. S. Willsky. Stochastic variational inference for Bayesian time series models. In *International Conference on Machine Learning*, 2014.

[14] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[15] S. Frühwirth-Schnatter. *Finite mixture and Markov switching models*. Springer Verlag, 2006.

[16] S. L. Scott. Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457):337–351, March 2002.

[17] M. J. Beale. *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. thesis, University College London, 2003.

[18] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, November 1999.

[19] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, 2006.

[20] B. T. Polyak and Y. Tsypkin. Pseudo-gradient adaptation and learning algorithms. *Automatics and Telemechanics*, 3:45–68, 1973.

[21] J. Gonzalez, Y. Low, and C. Guestrin. Residual splash for optimally parallelizing belief propagation. In *International Conference on Artificial Intelligence and Statistics*, 2009.

[22] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[23] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[24] ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, September 2012.

[25] M. M. Hoffman, O. J. Buske, J. Wang, Z. Weng, J. A. Bilmes, and W. S. Noble. Unsupervised pattern discovery in human chromatin structure through genomic segmentation. *Nature Methods*, 9:473–476, 2012.

[26] N. Day, A. Hemmaplardh, R. E. Thurman, J. A. Stamatoyannopoulos, and W. S. Noble. Unsupervised segmentation of continuous genomic data. *Bioinformatics*, 23(11):1424–1426, 2007.

[27] M. M. Hoffman, J. Ernst, S. P. Wilder, A. Kundaje, R. S. Harris, M. Libbrecht, B. Giardine, P. M. Ellenbogen, J. A. Bilmes, E. Birney, R. C. Hardison, M. Dunham, I. Kellis, and W. S. Noble. Integrative annotation of chromatin elements from encode data. *Nucleic Acids Research*, 41(2):827–841, 2013.