Scalable Clustering of Correlated Time Series using Expectation Propagation

Christopher Aicher
University of Washington
Department of Statistics
aicherc@uw.edu

Emily B. Fox
University of Washington
Department of Statistics
ebfox@uw.edu

ABSTRACT

We are interested in finding clusters of time series such that series within a cluster are correlated and series between clusters are independent. Existing Bayesian methods for inferring correlated clusters of time series either: (i) require conditioning on latent variables to decouple time series, but results in slow mixing or (ii) require calculating a collapsed likelihood, but with computation scaling cubically with the number of time series per cluster. To infer the latent cluster assignments efficiently, we consider approximate methods that trade exactness for scalability. Our main contribution is the development of an expectation propagation based approximation for the collapsed likelihood approach. Our empirical results on synthetic data show our methods scale linearly instead of cubically, while maintaining competitive accuracy.

1. INTRODUCTION

We are interested in finding clusters of time series such that series within a cluster are correlated and series between clusters are independent. We take motivation from a housing application analyzed by Ren et al. [10], though the methods are much more broadly applicable. In the housing application, the goal is to estimate house values at very fine spatial resolutions, such as the census tract level or finer. Housing trends vary over time and space, and the spatial structure is very heterogeneous where neighboring census tracts can behave quite differently. As a result, Ren et al. simply treat the house price processes within the set of census tracts as a collection of time series, ignoring spatial structure. However, the census tracts cannot be analyzed independently while still providing reasonable estimates of house value due to the scarcity of spatiotemporally localized house sales observations. To handle this data scarcity, Ren et al. proposed a method for discovering groups of *correlated* census tracts extending the correlated clustering model of Palla et al. [8] to time series data. Through discovering such a clustering of time series, one can improve estimates of local trends by sharing information via a form of multiple shrinkage.

In the housing application, the goal of inferring the clustering is to improve predictive performance. This type of clustering-of-time-series approach also proved useful in crime forecasting [1]. In other scenarios, the goal of clustering may be to produce an interpretable structure for understanding the relationships between time series. There is widespread demand for such time series clustering approaches.

Unfortunately, while the Bayesian model of Ren et al. [10] provides performance gains over alternative approaches, a significant limitation of the method is the complexity of the Bayesian inference procedure. In particular, inference of the cluster assignments of the individual time series presents a huge computational bottleneck: each single assignment update requires a likelihood computation with runtime scaling cubicly with the number of time series per cluster. This costly step has to be repeated *for each time series and each possible cluster assignment at each iteration* of the Bayesian inference algorithm. Unfortunately, due to the structure of the problem, there are no opportunities for sharing computations between steps.

More specifically, to perform Bayesian inference of the cluster assignments, a Gibbs sampler is used to iteratively draw the assignments and other parameters from the posterior. Standard implementations of Gibbs sampling are known to exhibit poor performance (slow mixing) when inferring a large number of parameters [13]. To overcome this slow mixing of the naive Gibbs sampler, Ren et al. [10] developed a collapsed Gibbs sampler that analytically marginalizes a large portion of the parameter space, but induces dependencies between series previously decoupled (via latent processes) in the uncollapsed model. Importantly, the resulting collapsed model still maintains tractable time series structure: a multivariate state space model per cluster. When resampling the cluster assignment for a given time series, a Kalman filter can be run to compute the likelihood of the series under a given cluster assignment. But, the Kalman filter has complexity scaling cubicly with the state dimension of the cluster. As a result, the resampling step, for just a single update to a given cluster assignment, has complexity $O(KN_{max}^3T)$, where K is the number of cluster, T is the number of time points, and N_{max} is the maximum cluster size. This computationally intensive step has to be repeated for each time series at each iteration. As a result, although the naive Gibbs sampler mixes slowly (i.e., takes many iterations), the collapsed Gibbs sampler has prohibitively slow runtimes for moderate to large cluster sizes,

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

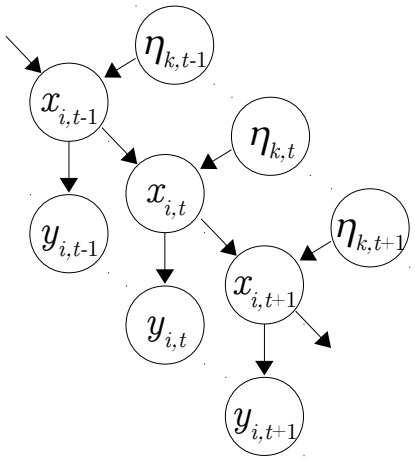

Figure 1: Graphical model for Eq. (4) for a single time series y_i with $z_i = k$. Conditioned on the set of z_i and η_k, each time series evolves independently. (The sharing of η_k between series in a cluster implicitly correlates the series.)

obviating the potential gains of collapsing. We demonstrate these effects in our experiments.

For tractable inference, we instead consider approximations to the collapsed likelihood that trade exactness for scalability. Existing methods for scaling Bayesian inference with the size of the data, such as stochastic variational inference [4] or Firefly MCMC [6], require independence between observations; however our model's observations are *dependent* within clusters after integrating out cluster latent variables. This is a common occurrence in models where collapsing emission parameters is common practice, like in latent Dirichlet allocation (LDA), or other mixture models with large parameter spaces. As such, the methods we develop in this paper are more broadly applicable than just to the specific dynamic model considered.

To handle the dependencies between observations, we develop two orthogonal methods: one based on *subsampling* and another based on *expectation propagation* [7]. We also present a combined method, *expectation propagation with subsampling*, that combines the insights of both. The three of methods all scale linearly in the number of time series per cluster. Our synthetic experiments find that both EP-based samplers mix as well as collapsed Gibbs in terms of performance versus number of iterations, while significantly outperforming collapsed Gibbs in terms of runtime.

The rest of the paper is organized as follows. We first present the model for correlated time series clusters. We then review naive and collapsed Gibbs samplers and present our approximate samplers. Finally, we analyze the trade-off between runtime and accuracy in synthetic data.

2. MODEL FOR TIME SERIES CLUSTERS

Let $y = \{y_i \in \mathbb{R}^T\}_{i=1}^N$ be a collection of N observed time series. Here, we assume that each series y_i is of length T, but our formulation can more generally apply to collections of time series of different lengths.

We additionally assume for each series i that y_i are noisy observations of a latent AR(1) process $x_i \in \mathbb{R}^T$.

$$\begin{aligned} x_{i,t} &= a_i x_{i,t-1} + \epsilon_{i,t} & \epsilon_{i,t} &\sim \mathcal{N}(0, \sigma_{i,t}^2) \\ y_{i,t} &= x_{i,t} + \nu_{i,t} & \nu_{i,t} &\sim \mathcal{N}(0, \sigma_{y_i}^2) \end{aligned} \tag{1}$$

where $a_i \in \mathbb{R}$ is the AR coefficient for x_i.

Each series y_i has a latent cluster assignment $z_i \in 1, \ldots, K$ such that the latent process noise ϵ is correlated within clusters and independent between clusters. To capture this, Ren et al. [10] propose

$$\epsilon_{i,t} = \lambda_i \eta_{z_i,t} + \tilde{\epsilon}_{i,t} \qquad \tilde{\epsilon}_{i,t} \sim \mathcal{N}(0, \sigma_x^2) \ , \tag{2}$$

where $\eta_{k,t} \sim \mathcal{N}(0,1)$ is the latent factor process for cluster k and $\lambda_i \in \mathbb{R}$ are the factor loadings. Therefore, the covariance matrix of $\epsilon_t = (\epsilon_{1,t}, \ldots, \epsilon_{K,t}$ sorted by cluster assignment is block diagonal, since

$$\mathrm{Cov}(\epsilon_{i,t}, \epsilon_{j,t} \mid \lambda, z) = \begin{cases} \lambda_i \lambda_j \sigma_{\eta_k}^2 + \sigma_x^2 \mathbf{1}_{i=j} & \text{if } z_i = z_j = k \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Combining Eq. (1) and (2), an equivalent representation for the series dynamics is

$$\begin{aligned} x_{i,t} &= a_i x_{i,t-1} + \lambda_i \eta_{z_i,t} + \epsilon_{i,t} \\ y_{i,t} &= x_{i,t} + \nu_{i,t} \ . \end{aligned} \tag{4}$$

The graphical model of the random variables y, x, z, η described in Eq. (4) is visualized in Figure 1.

Figure 2 is an example of synthetic data generated from the model. Note that series within each cluster do not necessarily follow a mean trend, but are instead *correlated*. This represents

Priors

Taking a Bayesian approach, we assign priors to our model parameters and latent variables.

Ren et al. [10] take a nonparameteric and hierarchical Bayesian approach. To focus our analysis on the effects of different likelihoods on sampling z, we treat the AR coefficients $a_{1:N}$, factor loadings $\lambda_{1:N}$, and noise variances σ_y^2, σ_x^2 as known. We additionally fix K and place a Dirichlet-multinomial prior over z for simplicity

$$z_i \mid p \sim \text{Multinomial}(1, p) \quad p \sim \text{Dirichlet}(\alpha) \ , \tag{5}$$

where $\alpha \in \mathbb{R}_+^k$ is the hyperparameter of the Dirichlet prior.

One can analytically marginalize the cluster weights p and compute the conditional distribution of a cluster assignment z_i given the other cluster assignments z_{-i}

$$\Pr(z_i = k \mid z_{-i}, \alpha) \propto (N_k - 1) + \alpha_k \ , \tag{6}$$

where N_k is the number of series in group k.

3. INFERENCE

We are interested in scalable methods of inference of the latent cluster assignments $z_{1:N}$.

We first present both naive and collapsed Gibbs sampling. Then we present our approximate methods based on subsampling and expectation propagation.

Although indefinitely repeating steps of these approximate samplers does not guarantee convergence to the exact posterior, they can be viewed as part of an adaptive MCMC scheme where these efficient but approximate steps are gradually faded out. The results of such an adaptive MCMC scheme converge to the exact posterior.

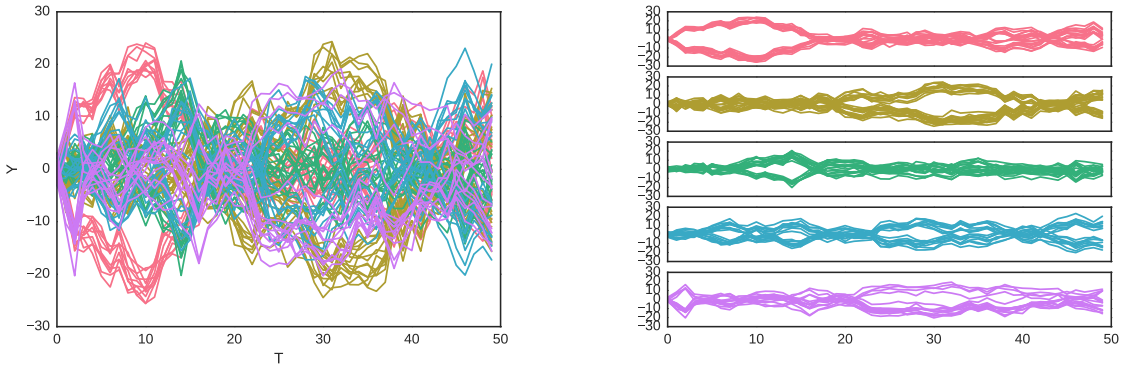

Figure 2: Synthetic data from the model: raw data (Left), separated by cluster (Right)

3.1 Gibbs Samplers

3.1.1 *Naive/Standard Gibbs*

A naive (or standard) Gibbs sampler draws samples (z, η) by iterating the following two steps

- For $i = 1 : N$,

$$z_i \sim \Pr(z_i \mid y, z_{-i}, \eta_{1:K}) \tag{7}$$

- For $k = 1 : K$,

$$\eta_k \sim \Pr(\eta_k \mid y, z_{1:N}, \eta_{-k}) \ . \tag{8}$$

Recall that to focus on the sampling of z, we fix all other model parameters that would otherwise be sampled.

We are interested in the computational cost of sampling z. Due to the conditional independence of z and y given η under out model, the full conditional for z_i in Eq. (7) simplifies to

$$\Pr(z_i = k \mid y, z_{-i}, \eta) \propto \Pr(z_i = k \mid z_{-i}) \Pr(y_i \mid \eta_k) \ , \tag{9}$$

which consists of a prior term and likelihood term. To calculate the full conditional, we must normalize the product of the prior and the likelihood for each cluster assignment $z_i = k$.

The prior term $Pr(z_i = k \mid z_{-i})$ is calculated from Eq. (6) and takes $O(N)$ time. The likelihood term

$$\ell_i^{\text{naive}}(z_i = k \mid \eta) = \Pr(y_i \mid \eta_k) \ , \tag{10}$$

can be calculated by running a Kalman filter on series y_i alone[2], with a slight modification to account for the fixed mean term $\lambda_i \eta_k$. For each cluster k in $1 : K$ we calculate

$$\ell_i^{\text{naive}}(z_i = k \mid \eta) = \prod_{t=1}^{T} \Pr(y_{i,t} \mid \eta_k, y_{i,1:t-1}) \ , \tag{11}$$

which has a runtime complexity of $O(T)$. Therefore, sampling z_i for each series i has a runtime complexity of $O(TK)$.

Altogether, sampling $z_{1:N}$ with naive Gibbs takes $O(TKN)$ time. This is linear in N. Unfortunately, the naive Gibbs sampler has slow mixing: A poor initialization z leads to poor estimates of η and vice-versa, requiring many iterations of the sampler to explore the posterior well.

3.1.2 *Collapsed Gibbs*

To improve the convergence of naive Gibbs, collapsed Gibbs samples the latent cluster assignments z from the conditional posterior after analytically *integrating out* η

$$z_i \sim \Pr(z_i \mid y, z_{-i}) \ . \tag{12}$$

Because the distribution of z_i is not conditioned on the sampled $\eta_{1:K}$, collapsed Gibbs (Eq. (12)) should mix in fewer iterations than naive Gibbs (Eq. (7)).

However, the computational cost of sampling z now requires collapsing out $\eta_{1:K}$ in the likelihood calculation.

The sampling distribution over z in Eq. (12) still factorizes into a prior and likelihood term

$$\Pr(z_i \mid y, z_{-i}) \propto \Pr(z_i \mid z_{-i}) \Pr(y_i \mid y_{-i}, z) \ , \tag{13}$$

but with a new 'collapsed' likelihood term that integrates out η.

$$\ell_i(z_i = k) = \Pr(y_i \mid y_{-i}, z) = \int \Pr(y_i \mid \eta_k) \Pr(\eta_k \mid y_{-i}, z_{-i}) \, d\eta_k \ . \tag{14}$$

The collapsed likelihood is the expected value of the naive likelihood with respect to the posterior of η_{z_i} given the observations and cluster assignments of other series y_{-i}, z_{-i}.

Figure 3 shows the dependencies induced by collapsing out η for one cluster.

Unfortunately, it is intractable to compute the integral in Eq. (14) as the posterior of $\eta_k \in \mathbb{R}^T$ is a full multivariate Gaussian. Direct integration would require $O(T^3)$ time.

One of the key results of Ren et al. [10] was to exploit the time series structure of the model (see Figure 3) to compute ℓ_i using a Kalman filter, but now on the collection of time series in a cluster. From the definition of conditional probability

$$\ell_i(z_i = k) = \Pr(y_i \mid y_{-i}, z) = \frac{\Pr(\{y_j : z_j = k\})}{\Pr(\{y_j : z_j = k \text{ and } j \neq i\})} \ . \tag{15}$$

For each potential cluster assignment $z_i = k$, the likelihood term now requires running the Kalman filter not only on the vector y_i but all series y_j in the same cluster $(z_j = z_i = k)$. As the Kalman filter scales cubically in the dimension of the state vector [2], if N_k is the number of series in cluster k, then the computational complexity to calculate $\ell_i(z_i = k)$ is $O(TN_k^3)$. Thus the running time for calculating the full

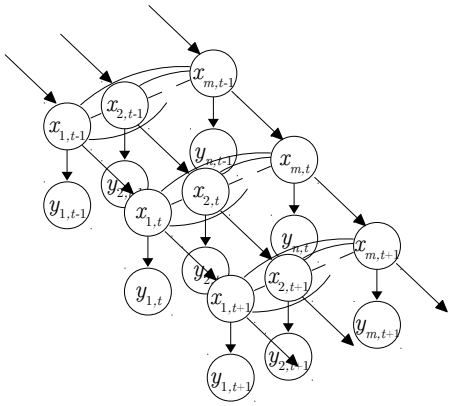

Figure 3: Graphical model for Eq. (4) where η is collapsed. Here we depict all series within a single cluster k, which in Figure 1 all shared the same η_k. The result of marginalizing η_k is the coupling between the $x_{i,t}$ depicted here. This is explicitly specified by the covariance in Eq. (3).

conditional pmf for z_i for each of its K possible values is $O(TK(\max N_k)^3)$.

Altogether, sampling the cluster assignments for all series $z_{1:N}$ with collapsed Gibbs takes $O(TKN(\max N_k)^3)$ time. Although collapsed Gibbs sampling improves upon the mixing, the cubic-scaling running time is intractable when the N_k is large and means fewer iterations of the algorithm given a fixed amount of time, limiting the theoretical gains of collapsing.

3.2 Approximate Samplers

We now consider approximations for the likelihood computation in collapsed Gibbs that trade exactness for significant computational speed-ups.

3.2.1 Subsampling

The problem with collapsed Gibbs sampling is its cubic scaling for large cluster sizes N_k. To reduce the complexity of calculating the collapsed likelihood $\ell_i(k)$ when N_k is large, one simple idea is to artificially reduce N_k by subsampling the data. For example, to assign a new series to a cluster, it may suffice to only consider a subset of observed series from each cluster. Visually, examine Figure 2 and imagine dropping half the time series in each cluster. One could probably still do reasonably well for most subsample halves. This idea is similar to ideas in sparse regression Gaussian process literature [9] and firefly MCMC[6], where we reduce the likelihood calculation complexity by dropping some likelihood terms.

For each potential cluster assignment $z_i = k$, instead of conditioning on all series in the cluster $\{y_j : z_j = k\}$, we condition on a random subsample of the series assigned to cluster k

$$\ell_i^{\text{sub}}(z_i) = \Pr(y_i \mid y_{\mathcal{J}}) = \frac{\Pr(y_i, y_{\mathcal{J}})}{\Pr(y_{\mathcal{J}})} \tag{16}$$

where $y_{\mathcal{J}} \subset \{y_j : z_j = z_i \text{ and } j \neq i\}$.

This subsampled likelihood approximation is equivalent to calculating

$$\ell_i^{\text{sub}}(z_i = k) = \int \Pr(y_i \mid \eta_k) \Pr(\eta_k \mid y_{\mathcal{J}}) \, d\eta_k \ , \tag{17}$$

where $\Pr(\eta_k \mid y_{\mathcal{J}})$ is the posterior for η_k condition on the subset of series $y_{\mathcal{J}}$.

The complexity for calculating ℓ_i^{sub} is the same as the collapsed Gibbs sampler, replacing N_k with the size of $y_{\mathcal{J}}$. In practice, we set an upper bound M for the size of our random subsample $y_{\mathcal{J}}$; therefore our total running time is $O(TKNM^3)$.

This likelihood and resulting posterior are equivalent to the exact approach if $y_{\mathcal{J}}$ is sufficient for $\{y_j : z_j = z_i\}$; however as $y_{\mathcal{J}}$ is chosen randomly, this is unlikely in general. Replacing ℓ_i with ℓ_i^{sub} in the collapsed Gibbs sampler leads to a different stationary distribution than the true posterior. Because ℓ_i^{sub} integrates over a posterior conditioned on less data (compare Eq. (17) and Eq. (14)), this new stationary distribution may be more diffuse than the exact posterior. The hope is that $y_{\mathcal{J}}$ is large enough to be 'approximately' sufficient for $\{y_j : z_j = z_i\}$, but small enough to be computational fast to calculate.

3.2.2 Expectation Propagation

Expectation propagation (EP) [7, 11] is an alternative method for reducing the complexity of calculating the collapsed likelihood.

Recall the collapsed likelihood can be represented as the integral over the posterior of η_{z_i} given y_{-i}, z_{-i}

$$\ell_i(z_i = k) = \int \Pr(y_i \mid \eta_k) \Pr(\eta_k \mid y_{-i}, z_{-i}) \, d\eta_k \ , \tag{18}$$

but is intractable to compute as the posterior of $\eta \in \mathbb{R}^{K \times T}$ has a full multivariate Gaussian density

$$\Pr(\eta \mid y_{-i}, z_{-i}) = \pi_{-i}(\eta) = \pi(\eta) \prod_{j \neq i} s_j(\eta) \tag{19}$$

where $\pi(\eta) = \mathcal{N}(\eta | 0, \text{diag}(\sigma_\eta^2))$ and s_j is the likelihood for η given y_j, z_j

$$s_j(\eta) = \Pr(y_j \mid \eta_{z_j}) \propto \mathcal{N}(\eta_{z_j} \mid \mu_j, \Sigma_j) \ , \tag{20}$$

which is a full multivariate Gaussian.

The EP idea is to approximate the likelihood terms s_j, such that the approximate posterior for η has the same form as its prior: a diagonal Gaussian. By approximating s_j with diagonal Gaussians, Eq. (18) becomes a tractable integral over a diagonal Gaussian posterior.

Let $t_j(\eta)$ be the diagonal Gaussian approximation of $s_j(\eta)$. Then define q_{-i} to be the approximate posterior of $\pi_{-i}(\eta)$

$$q_{-i}(\eta) \propto \pi(\eta) \prod_{j \neq i} t_j(\eta) \ . \tag{21}$$

Replacing $\pi_{-i}(\eta) = \Pr(\eta | y_{-i}, z_{-i}$ with $q_{-i}(\eta)$ in Eq. (18) gives us our EP approximation for the collapsed likelihood

$$\ell_i^{\text{EP}}(z_i) = \int \Pr(y_i \mid \eta, z_i) q_{-i}(\eta) \, d\eta \ . \tag{22}$$

The integral can be calculated efficiently using the Kalman filter by treating $q_{-i}(\eta)$ as the prior. This EP approximation ℓ_i^{EP} to the collapsed likelihood ℓ_i is good when the posterior $\pi_{-i} = \Pr(\eta \mid y_{-i}, z_{-i})$ is well approximated by $q_{-i}(\eta)$.

We now describe how to construct our likelihood approximations t_i for s_i. The standard EP update rule [7] is to select t_i to minimize the Kullback-Leibler divergence

$$t_i = \underset{\tilde{t}_i}{\operatorname{argmin}} \, \mathrm{KL}\left(s_i q_{-i} \;||\; \tilde{t}_i q_{-i}\right) \quad . \tag{23}$$

Since t_i is a diagonal Gaussian, minimizing the KL-divergence is equivalent to matching the marginal mean and variance of the tilted density $\tilde{q}_i(\eta) = \pi_{-i}(\eta)s_i(\eta)$. Therefore, we only need to calculate the marginal densities of $\tilde{q}_i$.

The marginal tilted density $\tilde{q}_i$ at time t can be efficiently calculated by exploiting the conditional independence of η and y given x (see Figure 1)

$$\tilde{q}_i(\eta_t) = \int \Pr(\eta_t|x_{i,t}, x_{i,t-1}) P_q(x_{i,t}, x_{i,t-1}|y_i) \; dx_{i,t} dx_{i,t-1} \quad , \tag{24}$$

where $P_q(x_{i,t}, x_{i,t-1}|y_i)$ is calculated using a Kalman smoother with $q_{-i}(\eta)$ as the prior for η and where $\Pr(\eta_t|x_{i,t}, x_{i,t-1})$ is

$$\Pr(\eta_t|x_{i,t}, x_{i,t-1}) = \mathcal{N}(\eta_t \mid (x_{i,t} - a_i x_{i,t-1})/\lambda_i, \sigma_x^2) \quad . \tag{25}$$

The runtime complexity of this algorithm depends on the complexity of its two steps: integrating over q_{-i} in Eq. (22) and updating t_j in Eq. (23).

For the first step, integrating out q_{-i} is equivalent to calculating the likelihood of the dynamical model (4) where η is distributed according to q_{-i}. Since the state vector is one dimensional, the Kalman filter takes $O(T)$ time for each cluster assignment $z_i = k$; thus the first step takes $O(TKN)$ time.

For updating t_j, each series i requires a one-dimensional Kalman smoother step and a simple bivariate Gaussian integral (Eq. (24)). Therefore the second step takes $O(TN)$ time.

Altogether, the EP-based approximate sampler takes $O(TKN)$ per iteration.

3.2.3 *Expectation Propagation with Subsampling*

We can combine the previous two approximation methods to obtain a third approximation for the collapsed likelihood $\ell_i(z_i)$.

EP simplifies the calculation by approximating complex likelihood terms s_j with a simpler form t_j. This is in contrast with the subsampling method, which makes no approximation on likelihood terms s_j, but only uses a random subset for computational tractability (ignoring the other terms). Our EP and subsampling approach combines both methods: it approximates most terms with the EP t_j, but selects a random subset to be treated exactly like in subsampling.

Let $\mathcal{J}$ be a subset of $\{1, \ldots, N\}\backslash\{i\}$. Then our 'EP with subsampling' approximation for the collapsed likelihood is

$$\ell_i^{\mathrm{EPsub}}(z_i) \propto \int \Pr(y_i \mid \eta, z_i)\pi(\eta) \prod_{j\in\mathcal{J}, j\neq i} s_j(\eta) \prod_{j\notin\mathcal{J}} t_j(\eta) \; d\eta \quad . \tag{26}$$

This integral is intractable to evaluate directly, but can be calculated using the same conditional probability trick as in our subsampling approach Eq. (16)

$$\ell_i^{\mathrm{EPsub}}(z_i) = \frac{P_{\mathcal{J}}(y_i, y_{\mathcal{J}})}{P_{\mathcal{J}}(y_{\mathcal{J}})} \quad , \tag{27}$$

but where $P_{\mathcal{J}}$ denotes the likelihood under the Kalman filter treating $q_{-\mathcal{J}}(\eta)$ as the prior for η.

$$q_{-\mathcal{J}}(\eta) \propto \pi(\eta) \prod_{j\notin\mathcal{J}, j\neq i} t_j(\eta) \quad . \tag{28}$$

We can view this as incorporating the missing terms from subsampling into the prior as if we applied Bayes rule twice: first update on $y_j \notin y_{\mathcal{J}}$, then update on $y_{\mathcal{J}}$, where we approximate the intermediate posterior of η with the diagonal Gaussian $q_{-\mathcal{J}}(\eta)$.

This has the same runtime complexity as the subsampled approximation, but with a little extra overhead for updating t_j and calculating $q_{-\mathcal{J}}$ as in EP.

4. EXPERIMENTS

To assess the computational complexity and cluster assignment accuracy of our sampling methods, we perform experiments on synthetic data from the model.

For simplicity, we consider synthetic data sets with $K = 5$ clusters, $T = 100$ data points per series, $a_i = 0.95$, $\sigma_x^2 = 1.0$, and $\sigma_y^2 = 1.0$. Finally, we set $\alpha >> K$ so that the K clusters would have roughly the same number of series. We treat these parameters as known to focus on how our approximations perform on inferring η and z.

We vary the number of series N and the factor loadings λ_i in our experiments. The number of series N determines the size of the data set. We set the factor loadings $\lambda_i = \pm\lambda_*$ with equal probability. The tuning parameter λ_* determines the signal-to-noise (SNR) ratio of the data set: when λ_* is large, the series are more strongly correlated.

The five sampling methods we compare are:

- `Naive Gibbs` - includes sampling $\eta_{1:K}$.
- `Collapsed Gibbs`
- `Subsampled` - our subsampled approximation with a max subsample size $M = 5$.
- `EP` - our EP approximation, initializing $t_j = \mathcal{N}(0, \mathrm{inf})$.
- `EP-Subsampled` - our combined EP and subsampling approximation with a max subsample size $M = 5$.

4.1 Running Time Complexity

For our first experiment, we compare the running time of each algorithm as a function of the data set size N, holding $\lambda_* = 1$ constant. Figure 4 plots the average running time per iteration (sampling z) as a function of dataset size. The error bars are one standard error replicated over 20 trials. From Figure 4, it is clear that collapsed Gibbs scales superlinearly, while the other four methods have linear scaling. Figure 4 shows that collapsed Gibbs is intractable for large data and is the motivation for considering faster samplers.

4.2 Approximation Accuracy

We measure the inference performance of the five sampling methods by testing how quickly the sampled cluster assignments $\hat{z}$ approach the true cluster assignments z. We initialize all five methods with the same random starting points and average the results over 20 trials with $N = 300$ time series each.

To compare $\hat{z}$ and z, we use normalized variation of information (NVI), an information-theoretic metric for distance between clusters [14]

$$\mathrm{NVI}(\hat{z}, z) = 1 - I(\hat{z}, z)/H(\hat{z}, z) \quad , \tag{29}$$

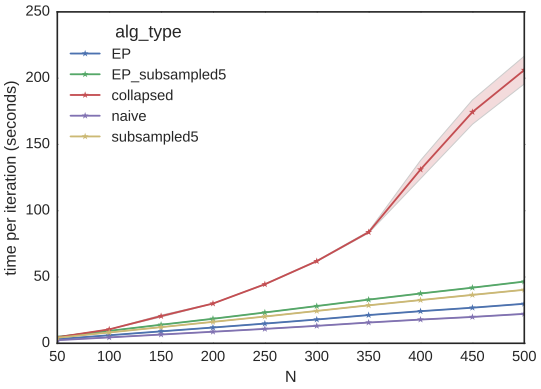

Figure 4: Per iteration runtime (seconds) vs number of series (N) for $K = 5$ groups

where I is mutual information and H is the joint entropy. The NVI between two clusters is minimized at 0 when the assignments are equal up to permutation and maximized at 1 when the mutual information between the two clustering is zero.

We consider two settings of λ_*: a lower SNR setting $\lambda_*^2 = 0.50$ and a higher SNR setting $\lambda_*^2 = 0.75$. Because we are assessing each method's ability to recover the true cluster assignments, we do not consider very low SNR settings.

Figure 5 depicts our results. The left column is the lower SNR setting of $\lambda_*^2 = 0.50$ and the right column is the higher SNR setting of $\lambda_*^2 = 0.75$. The first row of Figure 5 plots the simulated data broken down by true cluster assignments to provide a visualization of the challenge posed by the clustering task.

The second row of Figure 5 shows the NVI of the sampled cluster assignments to the true cluster assignments per iteration. In both SNR settings, `Collapsed Gibbs` and both EP-based methods converge quickly towards the true cluster assignment. The `Naive Gibbs` algorithm takes more iterations to converge due to its slower mixing. Finally, the `Subsampled` algorithm performs well in the higher SNR setting (right), but performs poorly in the lower SNR setting (left). The poor performance of `Subsampled` can be explained by two factors: (i) the lower SNR setting makes subsampling more challenging and (ii) its stationary distribution is more diffuse than the true distribution.

The third row of Figure 5 presents the same NVI results as the second row, but with the x-axis scaled by the average running time per iteration of each algorithm. This figure helps illustrate the trade-off between accuracy (y-axis) and time (x-axis). From third row, it is clear that the EP-based samplers outperform both `Naive Gibbs` (in accuracy) and `Collapsed Gibbs` (in scalability/speed).

Finally, the fourth row is the same as the third row, but zoomed-in on the lower right. We see that `EP` converges quickly in both settings, but `EP-Subsampled` eventually obtains a better NVI based on making a better approximation in each sampling step. As a result, we see that selecting `EP` versus `EP-Subsampled` represents an accuracy versus speed tradeoff, as expected. That said, even the slower (but more accurate) `EP-Subsampled` presents significant computational gains over the naive or collapsed samplers. One thing left to explore is the impact of these differences within the full MCMC where all the model parameters are resampled. In this context, the greater accuracy of `EP-Subsampled` could be important. Likewise, the performance of `Naive Gibbs` may be even worse relative to the comparison methods.

5. CONCLUSION AND FUTURE WORK

In this paper, we developed likelihood approximation methods to tractably infer the latent cluster assignments of correlated time series. We presented a model for multiple time series that produces correlated clusters. We showed that the existing collapsed Gibbs sampler is intractable for this task due to the cubic scaling of its collapsed likelihood calculation. We developed three approximations methods for approximating this likelihood that scale only linearly with the number of time series. This drastically reduces the runtime in large datasets since the call to resample cluster assignments is the majority of sampling steps and was previously the bottleneck to scalability of the model. All other sampling steps in [10] have negligible runtime in comparison.

We performed experiments on synthetic data focusing on learning the cluster assignments. We found that our subsampled approximation performs well in 'easy' settings where the posterior is concentrated, but performs poorly in 'hard' settings due to the diffuseness of its stationary distribution. On the other hand, our two EP-based approximations performed well in all settings having comparable performance per iteration with collapsed Gibbs, but scaling linearly instead of cubically with the cluster size.

The next step is to analyze this approximation in a fully Bayesian inference algorithm that learns all parameters.

Furthermore, although we applied our results using Gibbs sampling, our collapsed likelihood approximation can be extended to other approximate Bayesian inference methods such as stochastic variational inference. Additionally, we are interested in applying similar EP-approximations to collapsed likelihoods in other hierarchical models where the observations are dependent after integrating out latent variables.

6. ACKNOWLEDGMENTS

We would like to thank Nick Foti, You "Shirley" Ren and Alex Tank for helpful discussions.

This paper is based upon work supported by the NSF Career Award IIS-1350133.

7. REFERENCES

[1] S. Aldor-Noiman, L. Brown, E. Fox, and R. Stine. Spatio-temporal low count processes with application to violent crime events. *to appear in Statistica Sinica*, 2016.

[2] D. Barber, A. T. Cemgil, and S. Chiappa. Inference and estimation in probabilistic time series models. *Bayesian Time Series Models*, 2011.

[3] C. M. Bishop. Pattern recognition. *Machine Learning*, 2006.

[4] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1), 2013.

[5] T. W. Liao. Clustering of time series dataâĂŤa survey. *Pattern recognition*, 38(11), 2005.

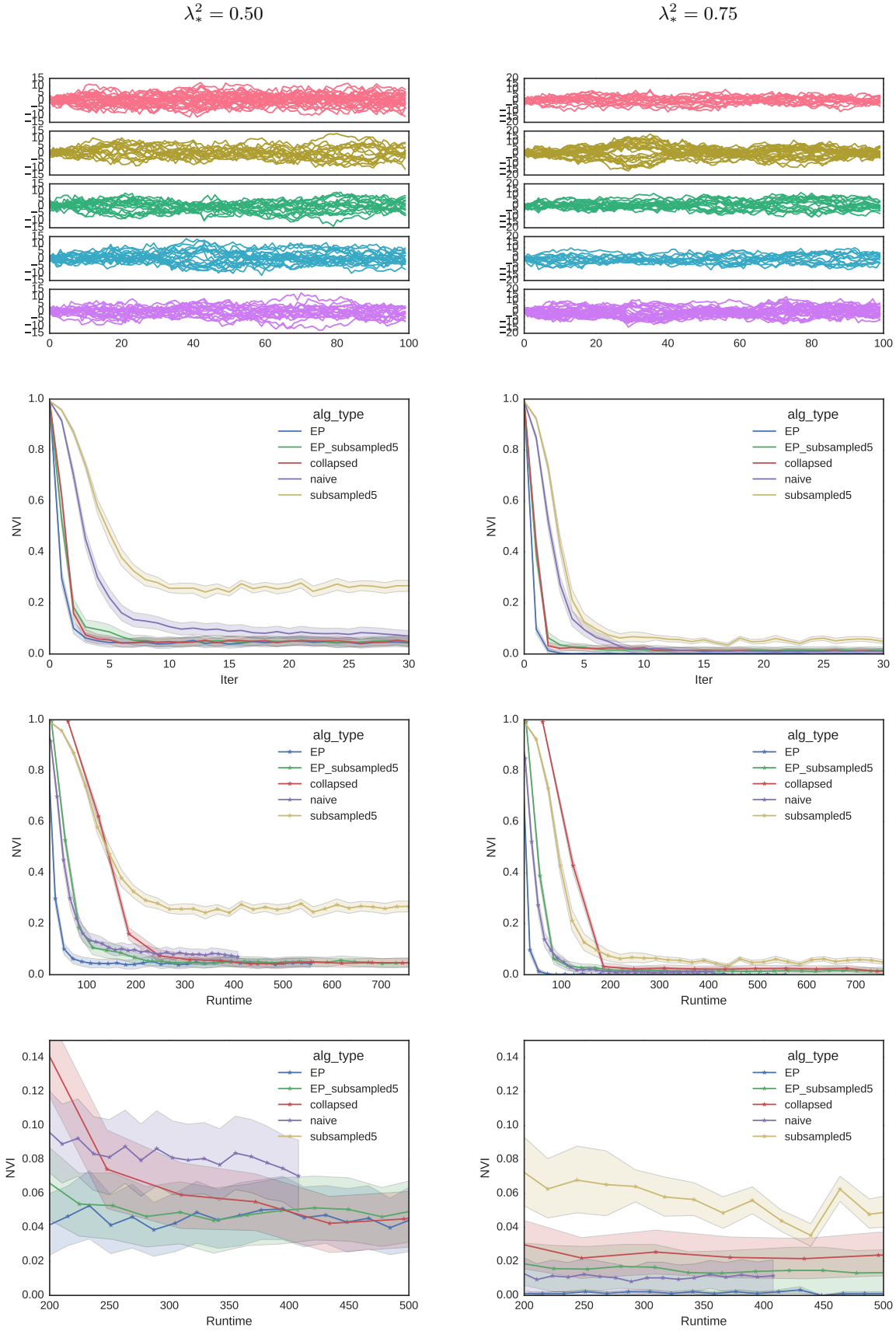

Figure 5: Comparison of clustering performance between `Naive Gibbs`, `Collapsed Gibbs`, `Subsampled`, `EP`, and `EP-Subsampled` on simulated data (*top row*, separated by true clustering) for $\lambda_*^2 = 0.50$ (*left*) and $\lambda_*^2 = 0.75$ (*right*). Comparisons show NVI versus iteration (*second row*), NVI versus runtime (*third row*), and a zoom in of the tail of the runtime plot (*fourth row*).

[6] D. Maclaurin and R. P. Adams. Firefly monte carlo: Exact mcmc with subsets of data. *arXiv preprint arXiv:1403.5693*, 2014.

[7] T. P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001.

[8] K. Palla, Z. Ghahramani, and D. A. Knowles. A nonparametric variable clustering model. In *Advances in Neural Information Processing Systems*, 2012.

[9] J. Quinonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6, 2005.

[10] Y. Ren, E. B. Fox, and A. Bruce. Achieving a hyperlocal housing price index: Overcoming data sparsity by bayesian dynamical modeling of multiple data streams. *arXiv preprint arXiv:1505.01164*, 2015.

[11] Y. W. Teh, L. Hasenclever, T. Lienart, S. Vollmer, S. Webb, B. Lakshminarayanan, and C. Blundell. Distributed bayesian learning with stochastic natural-gradient expectation propagation and the posterior server. *arXiv preprint arXiv:1512.09327*, 2015.

[12] Y. W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in neural information processing systems*, 2006.

[13] D. A. Van Dyk and T. Park. Partially collapsed gibbs samplers: Theory and methods. *Journal of the American Statistical Association*, 103(482), 2008.

[14] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11, 2010.

[15] Y. Xiong and D.-Y. Yeung. Time series clustering with arma mixtures. *Pattern Recognition*, 37(8), 2004.