Bayesian Conditional Density Filtering

Rajarshi Guhaniyogi, Shaan Qamar and David B. Dunson

October 16, 2014

Abstract

We propose a Conditional Density Filtering (C-DF) algorithm for efficient online Bayesian inference. C-DF adapts MCMC sampling to the online setting, sampling from approximations to conditional posterior distributions obtained by propagating surrogate conditional sufficient statistics (a function of data and parameter estimates) as new data arrive. These quantities eliminate the need to store or process the entire dataset simultaneously and offer a number of desirable features. Often, these include a reduction in memory requirements and runtime and improved mixing, along with state-of-the-art parameter inference and prediction. These improvements are demonstrated through several illustrative examples including an application to high dimensional compressed regression. Finally, we show that C-DF samples converge to the target posterior distribution asymptotically as sampling proceeds and more data arrives.

Keywords: Approximate MCMC; Big data; Density filtering; Dimension reduction; Streaming data; Sequential inference; Sequential Monte Carlo; Time series.

*Rajarshi Guhaniyogi, Assistant Professor, Department of Applied Math & Stat, SOE2, UC Santa Cruz, 1156 High Street, Santa Cruz, CA 95064 (E-mail: rguhaniy@ucsc.edu).
†Shaan Qamar, Ph.D. student, Department of Statistical Science, 022 Old Chemistry Building, Box 90251, Duke University, Durham, NC 27708-0251 (E-mail: shaan@stat.duke.edu).
‡David B. Dunson, Arts & Sciences Distinguished Professor, Department of Statistical Science, 218 Old Chemistry Building, Box 90251, Duke University, Durham, NC 27708-0251 (E-mail: dunson@duke.edu)
1 Introduction

Modern data are increasingly high dimensional, both in the number of observations $n$ and the number of predictors measured $p$. Statistical methods increasingly make use of low-dimensional structure assumptions to combat the curse of dimensionality (e.g., sparsity assumptions). When the number of observations is truly massive, however, data processing and computational complexity bottlenecks render many learning algorithms infeasible. Bayesian methods provide a natural probabilistic characterization of uncertainty in the parameters and in predictions, but there is a lack of scalable Bayesian inference algorithms having guarantees on accuracy. Here, sampling algorithms remain the gold standard, with the overwhelming emphasis being on Markov chain Monte Carlo (MCMC). State of the art performance is obtained in small to moderate sized problems, but MCMC fails to scale as the data and number of parameters grow (often due to the need to update many parameters with expensive likelihood evaluations).

MCMC can be extended to accommodate data collected over time by adapting the transition kernel $K_t$, and drawing a few samples at time $t$, so that samples converge in distribution to the joint posterior distribution $\pi_t$ in time (Yang and Dunson, 2013). Sequential MCMC (SMCMC) requires the full data to be stored, leading to storage and processing bottlenecks as more data arrive in time. When available, propagating sufficient statistics can help, although models with large parameters spaces often have sufficient statistics which are also high dimensional (e.g., linear regression). In addition, SMCMC often faces poor mixing requiring longer runs with additional storage and computation. A number of alternative strategies have been proposed for scaling MCMC to large datasets. One possibility is to parallelize computation within each MCMC iteration using GPUs or multicore architectures to free bottlenecks in updating unknowns specific to each sample and in calculating likelihoods (Medlar et al., 2013). Another possibility is to rely on Hamiltonian Monte Carlo with stochastic gradient methods used to approximate gradients with subsamples of the data (Welling and Teh, 2011). Ahn, Korattikara and Welling (2012) proposed a variation on this
theme. Korattikara, Chen and Welling (2013) instead use sequential hypothesis testing to choose a subsample to use in approximating the acceptance ratio in Metropolis-Hastings.

In simple conjugate models, such as Gaussian state-space models, efficient updating equations can be obtained using methods related to the Kalman filter. Assumed density filtering (ADF) was proposed (Lauritzen, 1992; Boyen et al., 1998; Opper et al., 1999) to extend this computational tractability to broader classes of models. ADF approximates the posterior distribution with a simple conjugate family, leading to approximate online posterior tracking. The predominant concern with this approach is the propagation of errors with each additional approximation to the posterior in time. Expectation-propagation (EP) (Minka et al., 2009; Minka, 2013) improves on ADF through additional iterative refinements, but the approximation is limited to the class of assumed densities and has no convergence guarantees. Moreover in fairly standard settings, arbitrary approximation to the posterior through an assumed density severely underestimates parametric and predictive uncertainties. A parallel literature on online variational approximations (Wang et al., 2011; Hoffman et al., 2010) focus primarily on improving batch inferences by feeding in data sequentially. Broderick et al. (2013) recently proposed a streaming variational Bayes algorithm, while Hoffman et al. (2012) combined stochastic approximation with variational inference for better scaling. However, variational methods rely on a factorized form of the posterior that typically fails to capture dependence in the joint posterior and severely underestimates uncertainty. Recent attempts to design careful online variational approximations (Luts et al., 2014), though successful in accurately estimating marginal densities, are limited to specific models and no theoretical guarantees on accuracy are established except for stylized cases.

Sequential Monte Carlo (SMC) (Chopin, 2002; Arulampalam, 2002; Lopes et al., 2010) is a popular technique for online Bayesian inference that relies on propagating and resampling samples in time. Unfortunately, it is difficult to scale SMC to problems involving large \( n \) and \( p \) due to the need to employ very large numbers of particles to obtain adequate approximations and prevent particle degeneracy. The latter is addressed through rejuvenation steps using
all the data (or sufficient statistics), which becomes expensive in an online setting. One
could potentially rejuvenate particles only at earlier time points, but this may not protect
against degeneracy for models involving many parameters. More recent particle learning
(PL) algorithms (Carvalho et al., 2010) reduce degeneracy for the dynamic linear model,
with satisfactory density estimates for parameters – but they too require propagating a large
number of particles, which significantly adds to the per-iteration computational complexity.

We propose a new class of Conditional Density Filtering (C-DF) algorithms that extend
MCMC sampling to streaming data. Sampling in C-DF proceeds by drawing from conditional
posterior distributions, but instead of conditioning on conditional sufficient statistics (CSS)
(Johannes et al., 2010; Carvalho et al., 2010), C-DF conditions on surrogate conditional
sufficient statistics (SCSS) using sequential point estimates for parameters along with the
data observed. This eliminates the need to store the data in time (process the entire dataset
at once), and leads to an approximation of the conditional distributions that produce samples
from the correct target posterior asymptotically. The C-DF algorithm is demonstrated to be
highly versatile and efficient across a variety of settings, with SCSS enabling online sampling
of parameters with dramatic reductions in the memory and per-iteration computational
requirements.

Section 2 introduces the C-DF algorithm in generality, along with definitions, assump-
tions, and a description of how to identify updating quantities of interest. Section 3 demon-
strates approximate online inference using C-DF in the context of several illustrating exam-
pies. Section 3.1 applies C-DF to linear regression and the one-way Anova model. More
sophisticated model settings are considered in Sections 3.2, namely extensions of the C-DF
algorithm to a dynamic linear model and binary regression using the probit model. Here, we
investigate the performance of C-DF in settings with an increasing parameter space. The C-
DF algorithm is also applied to a Poisson mixed effects model to update the parameters for a
variational approximation to the posterior distribution. We discuss various computational,
storage and mixing advantages of our method over state-of-the-art competitors in each.
Section 4 presents a detailed implementation of the C-DF algorithm for high dimensional compressed regression. We report state-of-the-art inferential and predictive performance across extensive simulation experiments as well as for real data studies in Section 5. Section 6 presents a finite-sample error bound for approximate MCMC kernels and establishes the asymptotic convergence guarantee for the proposed C-DF algorithm. Section 7 concludes with a discussion of extensions and future work. Proofs and additional figures pertaining to Section 3 appear in Appendix A and B, respectively.

2 Conditional density filtering

Define \( \Theta = (\theta_1, \theta_2, \ldots, \theta_k) \) as the collection of unknown parameters in probability model \( P(Y|\Theta) \) and \( Y \in \mathcal{Y} \), with \( \theta_j \in \Psi_j \), and \( \Psi_j \) denoting an arbitrary sample space (e.g., a subset of \( \mathbb{R}^p \)). Data \( D_t \) denotes the data observed at time \( t \), while \( D^{(t)} = \{ D_s, s = 1, \ldots, t \} \) defines the collection of data observed through time \( t \). Below, \( \theta_{-j} = \Theta \setminus \theta_j \) and let \( \theta_{-j} = (\theta_{-j,1}, \theta_{-j,2}) \) for each \( j = 1, \ldots, k \). Either of \( \theta_{-j,1} \) or \( \theta_{-j,2} \) is allowed to be a null set.

2.1 Surrogate conditional sufficient statistics

Definition: \( S^{(t)}_j \) is defined to be a conditional sufficient statistic (CSS) for \( \theta_j \) at time \( t \) if \( \theta_j \perp D^{(t)} | \theta_{-j,1}, S^{(t)}_j \). Suppose \( \theta_j | \theta_{-j}, D_t \leq \theta_j | \theta_{-j,1}, h(D_t, \theta_{-j,2}) \). Then it is easy to show that for known functions \( f, h \), \( S^{(t)}_j = f(h(D_1, \theta_{-j,2}), \ldots, h(D_t, \theta_{-j,2})) \). This satisfies \( \theta_j | \theta_{-j}, D^{(t)} \leq \theta_j | \theta_{-j,1}, S^{(t)}_j \), or equivalently \( \theta_j \perp D^{(t)} | \theta_{-j,1}, S^{(t)}_j \).

Because \( S^{(t)}_j \) depends explicitly on \( \theta_{-j,2} \), its value changes whenever new samples are drawn for this collection of parameters. This necessitates storing entire data \( D^{(t)} \) or available sufficient statistics. The following issues inhibit efficient online sampling: (1) sufficient statistics, when they exist, often scale poorly with the size of the parameter-space. This often creates a significant storage overhead when multiple copies of these data structures need to be created on a computer network (e.g., for ensemble gradient-descent based procedures); and (2) in
the context of sampling, updating $S_j^{(t)}$, $j = 1, \ldots, k$, for every iteration at time $t$ causes an immense per-iteration computational bottleneck. To address these challenges, we propose surrogate conditional sufficient statistics (SCSS) as a means to approximate full conditional distribution $\theta_j | \theta_{-j}, D^{(t)}$ at time $t$.

**Definition:** Suppose $\theta_j | \theta_{-j}, D_t \overset{\Delta}{=} \theta_j | \theta_{-j,1}, h(D_t, \hat{\theta}_{-j,2})$. For known functions $g, h$, define $C_j^{(t)} = g(C_j^{(t-1)}, h(D_t, \hat{\theta}_{-j,2}))$ as the surrogate conditional sufficient statistic for $\theta_j$, with $\hat{\theta}_{-j,2}$ being a consistent estimator of $\theta_{-j,2}$ at time $t$. Then $\theta_j | \theta_{-j,1}, C_j^{(t)}$ is the C-DF approximation to $\theta_j | \theta_{-j}, D^{(t)}$.

If the full conditional for $\theta_j$ admits a surrogate quantity $C_j^{(t)}$, approximate sampling via C-DF proceeds by drawing $\tilde{\theta}_j \sim \tilde{\pi}_j(\cdot | \theta_{-j,1}, C_j^{(t)})$. Crucially, $C_j^{(t)}$ depends only on point estimates for $\theta_{-j,2}$ which remain fixed while drawing samples from the approximate full conditional distribution. This avoids having to update potentially expensive functionals $h(D^{(t)}, \theta_{-j,2})$ (i.e., CSS) at every MCMC iteration. If the approximating full conditionals are conjugate, then sampling proceeds in a Gibbs-like fashion. Otherwise, assuming a proposal distribution $p_j(\cdot)$ for $\theta_j$, draws from $\tilde{\pi}_j(\theta_j | \theta_{-j,1}, C_j^{(t)})$ may be obtained via a Metropolis step with acceptance probability

$$\alpha_j(\theta' | \theta) = \frac{\tilde{\pi}_j(\theta' | \theta_{-j,1}, C_j^{(t)})/p_j(\theta')}{\tilde{\pi}_j(\theta | \theta_{-j,1}, C_j^{(t)})/p_j(\theta)}.$$  

(1)

Section 3.1 provides examples of the former, and Section 3.2.1 of the latter. In both cases, draws using the C-DF approximation to the conditional distributions have the correct asymptotic stationary distribution (see Section 6). Another scenario arises when a non-conjugate full conditional fails to admit SCSS (and therefore also SS or CSS) as in the case of the Poisson regression example in Section 3.2.3. A variational approximation to the joint posterior admits SCSS, and we demonstrate using the C-DF algorithm when a full conditional $\pi_j(\theta_j | -)$ is replaced by an approximating kernel $q_j(\theta_j | \theta_{-j,1}, C_j^{(t)})$ at time $t$. Propagating SCSS associated with the latter allows us to obtain approximate online inference using vari-
ation approximations to the posterior distribution. Here, the limiting distribution for draws sampled from \( q_j(\theta_j | \theta_{-j,1}, C_j^{(t)}) \) are not guaranteed to have the correct asymptotic stationary distribution.

### 2.2 The C-DF algorithm

Define a fixed grouping of parameters \( \{\theta_j, j = 1, \ldots, p\} \) into sets \( \{\mathcal{G}_l, l = 1, \ldots, k\} \) satisfying \( \mathcal{G}_l \cap \mathcal{G}_{l'} = \emptyset, l \neq l' \), and \( \sum_l I(j \in \mathcal{G}_l) = 1, j = 1, \ldots, k \). The C-DF algorithm alternates between obtaining consistent estimates for \( \Theta_l = \{\theta_{j'}, j' \notin \mathcal{G}_l\} \) and sampling from approximate full conditionals for a group of parameters \( \{\theta^l_j : j \in \mathcal{G}_l, \quad C_j^{(t)} = f_j(\hat{\Theta}_l), \quad \Theta_l = \Theta \setminus \Theta_{G_l}\} \).

By default, parameter estimates are obtained by averaging over \( S \) draws from the approximating full-conditionals (in cases where closed-form expression for the conditional mean, say, is available then this is used instead). This leads to efficient updating equations for \( C_j^{(t)} \) based on \( C_j^{(t-1)} \) and incoming data \( D_t \), resulting in scalable parameter inference with sampling-based approximations to the posterior distribution.

Conditional independence assumptions for a model often suggest natural parameter partitions, though alternate partitions may be more suitable for specific tasks. For streaming data, these sets are identified to maximize computational and storage gains. In some applications, one may be interested in conditional inference, updating only estimates for “nuisance” parameter sets. Examples of various partitioning schemes are presented in the context of several illustrating examples in Section 3. A general outline for the C-DF algorithm is given in Algorithm 1.

### 3 Illustrating examples using C-DF

This section demonstrates approximate online inference using C-DF in the context of several illustrating examples. Section 3.1 applies C-DF to linear regression and the one-way Anova model. The choice of parameter partitions and detailed steps of the C-DF algorithm for
Algorithm 1 A sketch of the C-DF algorithm for approximate MCMC

Input: (1) Data shard $D_t$; (2) Index sets $\{\mathcal{G}_l\}$ for parameter partition; and (3) $C_j^{(t-1)}$ for every $j = 1,\ldots,k$. Require that $\sum_l I(j \in \mathcal{G}_l) = 1$ and $\mathcal{G}_l \cap \mathcal{G}_{l'} = \emptyset$, $l \neq l'$.

Output: SCSS $C_j^{(t)}$, and approximate samples $\tilde{\theta}_j^{(1)},\ldots,\tilde{\theta}_j^{(S)}$ for every $j = 1,\ldots,k$.

1: function CDF.SAMPLE($D_t, C_j^{(t-1)}$, $\mathcal{G}_l$)
2:     for $l = 1 : L$ do
3:         for $s = 1 : S$ do
4:             for $j \in \mathcal{G}_l$ do
5:                 Define $\Theta_{\mathcal{G}_l} = \{\theta_j, j \in \mathcal{G}_l\}$ and $\Theta_{\mathcal{G}_l}^{(j)} = \Theta_{\mathcal{G}_l} \setminus \theta_j$
6:                 Sample $\tilde{\theta}_j^{(s)} \sim \pi_j(\cdot|\Theta_{\mathcal{G}_l}^{(j)}, C_j^{(t)})$ \[ or $\tilde{\theta}_j^{(s)} \sim q_j(\cdot|\Theta_{\mathcal{G}_l}^{(j)}, C_j^{(t)})$ \]
7:             end for
8:         end for
9:      end for
10:      For each $j \in \mathcal{G}_l$: Set $\hat{\theta}_j \leftarrow \text{mean}(\tilde{\theta}_j^{(1:S)})$,
11:      For each $j \in \mathcal{G}_l$: Store $\tilde{\theta}_j^{(1)},\ldots,\tilde{\theta}_j^{(S)}$ as approximate posterior samples
12:     For each $m \neq l$ and $j' \in \mathcal{G}_m$: Update $C_{j'}^{(t)} \leftarrow g(C_{j'}^{(t-1)}, h(D_t, \hat{\Theta}_m), \Theta_m = \Theta \setminus \Theta_{\mathcal{G}_m}$
13:  end for
14: end function

these simpler models serve to build intuition. Here, the conjugate conditional distributions are parameterized in terms of low-dimensional (or univariate) sufficient statistics, hence there is no significant computational benefit to using C-DF for online inference. Instead, our main focus is assessing the accuracy of the C-DF approximation in these simple examples. More sophisticated model settings are considered in Sections 3.2, namely extensions of the C-DF algorithm to a dynamic linear model and binary regression using the probit model. Here, we investigate the performance of C-DF in settings with an increasing parameter space (e.g., data augmentation using latent variables for the probit regression). The C-DF algorithm is also used in the context of a Poisson mixed effects model to update the parameters for a variational approximation to the posterior distribution. In particular, updates to the variational parameters admit surrogate statistics which are propagated to yield approximate online inference. The examples in Section 3.2 do not satisfy conditions for which we derive asymptotic guarantees for samples drawn from the set of approximating (C-DF) full conditionals (see Section 6). In particular, theoretical accuracy of the C-DF approximation is ensured only
for finite parameter settings where conditional distributions admit surrogate statistics and may be sampled from directly or using a Metropolis step (e.g., examples in Sections 3.1, 4.1). Where applicable, the similarity between the full conditional distribution using sequential MCMC (SMCMC), \( \pi_t(\theta_j|-) \), and approximate C-DF full conditional, \( \tilde{\pi}_t(\theta_j|-) \), is measured as

\[
\text{Accuracy} = 1 - \frac{1}{2} \int |\pi_t(\theta_j) - \tilde{\pi}_t(\theta_j)| \ d\theta_j.
\] (2)

As defined, accuracy ranges between 0 and 1, with larger values implying better agreement between the two distributions. Accuracy plots are all given in Appendix B.

Fixing notation, data \( D_t \) denotes the data observed at time \( t \), while \( D^{(t)} = \{D_s, \ s = 1, \ldots, t\} \) defines the collection of data observed through time \( t \). Where appropriate, \( D_t = (X^t, y^t) \) with \( X^t = (x_{t1}, \ldots, x_{tn})' \) and \( y^t = (y_{t1}, \ldots, y_{tn})' \). Shards of a fixed size arrive sequentially over a \( T = 1000 \) time horizon, and 500 MCMC iterations are drawn to approximate the corresponding posterior distribution \( \pi_t \) at every time point \( t = 1, \ldots, T \). We report mean squared estimation accuracy for model parameters of interest, in addition to mean squared prediction error (MSPE), length and coverage of 95% predictive intervals where applicable. Results are averaged over 10 independent replications with associated standard errors appearing as subscripts in Tables. Plots showing kernel density estimates for marginal posterior densities for representative model parameters at various time points are also provided.

### 3.1 Motivating examples

#### 3.1.1 Linear regression

Consider the simple linear regression model for a \( p \)-dimensional predictor vector,

\[
y^t = X^t \beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I_n).
\] (3)
A standard Bayesian analysis proceeds with conjugate priors $\beta \sim N(0, I_p), \sigma^2 \sim IG(a, b)$, and having Gibbs full conditionals given by,

$$
\sigma^2|\beta, D^{(t)} \sim IG(\alpha', \beta'), \quad \text{with} \quad \alpha' = a + \frac{tn}{2}, \quad \beta' = b + \frac{S_t^{YY} - 2\beta'S_t^{XY} + \beta' S_t^{XX} \beta}{2} \\
\beta|\sigma^2, D^{(t)} \sim N(\mu_t, \Sigma_t), \quad \text{with} \quad \Sigma_t = (S_t^{XX} / \sigma^2 + I_p)^{-1}, \quad \mu_t = \Sigma_t S_t^{XY} / \sigma^2.
$$

These distributions are specified in terms of sufficient statistics $S_t^{XX} = S_{t-1}^{XX} + X^t X^t$, $S_t^{XY} = S_{t-1}^{YY} + X^t y^t$, and $S_t^{YY} = S_{t-1}^{YY} + y^t y^t$, enabling online inference using SMCMC. A C-DF implementation begins by defining a partition over the model parameters, and here we set $\Theta_{G_1} = \{\beta\}$ and $\Theta_{G_2} = \{\sigma^2\}$. Approximate sampling via C-DF then proceeds as:

1. Observe data $D_t$ at time $t$. If $t = 1$ initialize all parameters at some default values (e.g., $\beta = 0, \sigma^2 = 1$ assuming a centered and scaled response); otherwise set $\hat{\sigma}^2_t \leftarrow \hat{\sigma}^2_{t-1}$;

2. Update surrogate CSS for $\beta$, $C_{1,1}^t = C_{1,1}^{t-1} + X^t X^t / \hat{\sigma}^2_t$, $C_{1,2}^t = C_{1,2}^{t-1} + X^t y^t / \hat{\sigma}^2_t$;

3. Define $C_1^t = \{C_{1,1}^t, C_{1,2}^t\}$ and draw $S$ samples from the approximate Gibbs full conditional $\beta|\hat{\sigma}^2_t, C_1^{(t)} \sim N(\hat{\mu}_t, \hat{\Sigma}_t)$ ($\hat{\Sigma}_t = (C_{1,1}^t + I_p)^{-1}, \hat{\mu}_t = \hat{\Sigma}_t C_{1,2}^t$), and set $\hat{\beta}_t \leftarrow \text{mean}(\beta^{(1:S)})$ (or use the analytical expression for the posterior mean);

4. Update surrogate CSS for $\sigma^2$, $C_{2,1}^t = C_{2,1}^{t-1} + \hat{\beta}_t^t X^t X^t \hat{\beta}_t$, $C_{2,2}^t = C_{2,2}^{t-1} + \hat{\beta}_t^t X^t y^t$;

5. Define $C_2^t = \{C_{2,1}^t, C_{2,2}^t\}$ and draw $S$ samples from the approximate Gibbs full conditional $\sigma^2|\hat{\beta}_t, C_2^{(t)} \sim IG(\alpha', \beta' + (S_t^{YY} - 2C_{2,2}^t + C_{2,1}^t)/2)$, and set $\hat{\sigma}^2_t \leftarrow \text{mean}(\sigma^{2(1:S)})$ (or use the analytical expression for the posterior mean).

We compared the hold-out predictive performance and inference on model parameters $\beta, \sigma^2$ using SMCMC and C-DF. Data is generated using predictors drawn from $U(0,1)$, with true parameters $\beta_0 = (1, 0.5, 0.25, -1, 0.75)$ and $\sigma_0^2 = 25$. For both methods, density estimates for parameters are shown at $t = 200, 500$, with accuracy comparisons validating that approximate draws using the C-DF algorithm converge to the true stationary distribution.
in time. Excellent parameter MSE and coverage using the C-DF algorithm is reported in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Avg. coverage $\beta$</th>
<th>Length</th>
<th>Time (sec)</th>
<th>MSE = $\sum_{j=1}^{p}(\hat{\beta}_t - \beta_0)^2/p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>1.0</td>
<td>0.600,01</td>
<td>95.4,12</td>
<td>$0.27_{0.001}$, $0.15_{0.001}$, $0.06_{0.001}$</td>
</tr>
<tr>
<td>SMCMC</td>
<td>1.0</td>
<td>0.600,01</td>
<td>119.4,64</td>
<td>$0.12_{0.001}$, $0.08_{0.001}$, $0.04_{0.001}$</td>
</tr>
</tbody>
</table>

Table 1: Inferential performance for C-DF and SMCMC for parameters of interest. Coverage and length are based on 95% credible intervals and is averaged over all the $\beta_j$’s ($j = 1, \ldots, 5$) and all time points and over 10 independent replications. We report the time taken to produce 500 MCMC samples with the arrival of each data shard. MSE along with associated standard errors are reported at different time points.

Figure 1: Kernel density estimates for posterior draws using SMCMC and the C-DF algorithm at $t = 200, 500$. Shown from left to right are plots of model parameters $\beta_1$, $\beta_4$, and $\sigma^2$, respectively.

### 3.1.2 One-way Anova

Consider the one-way Anova model with a fixed number treatment groups, $k$,

$$y_{ij} = \zeta_i + \epsilon_{ij}, \quad \zeta_i \overset{iid}{\sim} N(\mu, \tau^2), \quad \epsilon_{ij} \sim N(0, \sigma^2), \quad (4)$$

with $\pi(\mu) \propto 1$, $\tau^2 \sim IG(a, b)$, and $\sigma^2 \sim IG(\alpha, \beta)$. Sufficient statistics can be propagated yielding an efficient SMCMC sampler. Having observed a new data shard at time $t$, group-specific sufficient statistics $S_i^t$ are updated as $S_i^t = S_i^{t-1} + \sum_{j=1}^{n} y_{ij}^t$ and $S_i^{2(t)} = S_i^{2(t-1)} +$
\[ \sum_{i=1}^{k} ||\mathbf{y}_i||^2, \quad i = 1, \ldots, k. \] 
At time \( t \), inference proceeds with draws from the following Gibbs full conditionals:

\[ 
\begin{align*}
\zeta_i|\sigma, \mu, \tau, \mathbf{y} &\sim N\left( \frac{\tau^2 S_i^t + \sigma^2 \mu}{n \tau^2 + \sigma^2}, \frac{\tau^2 \sigma^2}{n \tau^2 + \sigma^2} \right), \quad \sigma^2|\zeta, \mathbf{y} \sim IG\left( \alpha + \frac{nkt}{2}, \beta + \frac{\sum_{i=1}^{k} (S_i^{2(t)} - 2\zeta_i S_i^t + nt\zeta_i^2)}{2} \right) \\
\mu|\zeta, \tau &\sim N\left( \frac{\sum_{i=1}^{k} \zeta_i}{k}, \frac{\tau^2}{k} \right), \\
\tau^2|\zeta, \mu &\sim IG\left( a + \frac{k}{2}, b + \frac{\sum_{i=1}^{k} (\zeta_i - \mu)^2}{2} \right). 
\end{align*} 
\] (5)

Using the hierarchical structure of model (4), C-DF partitions the parameters as \( \Theta_{g_1} = \{\zeta, \sigma^2\} \), and \( \Theta_{g_2} = \{\mu, \tau^2\} \). The modified full conditionals are defined in terms of surrogate quantities, as well as the previously defined group-specific sufficient statistics. Approximate inference using C-DF proceeds as follows:

1. Observe data \( \mathbf{y}_1^t, \ldots, \mathbf{y}_k^t \) at time \( t \). If \( t = 1 \), set \( \zeta_i = 0, \sigma = \bar{s}(\text{vec} (\mathbf{y}_1^t, \ldots, \mathbf{y}_k^t)), \mu = 0, \tau = 1 \). Otherwise, set \( \hat{\mu}_t \leftarrow \hat{\mu}_{t-1}, \hat{\tau}_t \leftarrow \hat{\tau}_{t-1} \);

2. Update surrogate statistic \( C_1^t \) component-wise as \( C_{1i}^t \leftarrow C_{1i}^{t-1} + \hat{\tau}_i S_i^t, \quad i = 1, \ldots, k; \)

3. For \( s = 1, \ldots, S \): (a) draw \( \zeta_i|\sigma, \hat{\mu}_t, \hat{\tau}_t, C_1^t, \quad i = 1, \ldots, k; \) (b) draw \( \sigma^2|\zeta, \mathbf{y} \). The C-DF full conditional for group-specific means are given by

\[ 
\zeta_i|\sigma, \hat{\mu}_t, \hat{\tau}_t, C_1^t \sim N\left( \frac{C_{1i}^t + \sigma^2 \hat{\mu}_t}{n \hat{\tau}_i^2 + \sigma^2}, \frac{\hat{\tau}_i^2 \sigma^2}{n \hat{\tau}_i^2 + \sigma^2} \right), \quad i = 1, \ldots, k; 
\] (6)

4. Set \( \hat{\zeta}_i \leftarrow \text{mean}(\zeta_i^{(1:S)}), \quad i = 1, \ldots, k, \) and \( \hat{\sigma}^2 \leftarrow \sigma^2^{(1:S)}; \) using these values, update surrogate statistic \( C_{2,1}^t \leftarrow C_{2,1}^{t-1} + ||\hat{\zeta}||^2, \quad C_{2,2}^t \leftarrow C_{2,2}^{t-1} + \sum_{i=1}^{k} \hat{\zeta}_i \). Let \( C_2^t = (C_{2,1}^t, C_{2,2}^t); \)

5. For \( s = 1, \ldots, S \): sample from approximate full conditionals \( \mu|\tau, C_2^t \sim N\left( C_{2,1}^t/k, \tau^2/k \right) \) and \( \tau^2|\mu, C_2^t \sim IG\left( a+k/2, b + (C_{2,1}^t - 2\mu C_{2,2}^t + k\mu^2)/2 \right) \) defined in (5);

6. Finally, set \( \hat{\mu}_t \leftarrow \text{mean}(\mu^{(1:S)}) \) and \( \hat{\tau}_t \leftarrow \text{mean}(\tau^{(1:S)}). \)

ADF is added as an additional competitor for this example. Here, the joint posterior over \( \theta = (\zeta_1, \ldots, \zeta_k, \log(\sigma^2)) \) is approximated in time assuming a multivariate-normal density.
\( \tilde{\pi}_t(\theta) \sim N(\mu_t, \Sigma_t) \). To begin, we integrate over hyper-parameters \( \mu, \tau^2 \) to obtain marginal prior \( \pi(\zeta, \log(\sigma^2)) \). For \( t > 1 \), the approximate posterior at time \((t-1)\) becomes the prior at \( t \), and parameters \( \mu_t, \Sigma_t \) are updated in the sense of McCormick et. al (2012). In particular, \( \Sigma_t = (\nabla^2 \ell(\mu_{t-1}))^{-1} \) and \( \mu_t = \mu_{t-1} + \Sigma_{t-1} \nabla \ell(\mu_{t-1}) \), with \( \ell(\theta) = \log(p(y|\theta)\pi(\theta)) \).

Simulated data for this example is generated according to model (4) with parameters \( \zeta_i \text{ iid } \sim N(4, \tau^2 = 0.01) \) and \( \sigma^2 = 100 \). Kernel density estimates for posterior draws using SMCMC and the C-DF algorithm are very similar as depicted in Figure 2 at \( t = 200, 500 \). The C-DF approximation results in steadily increasing accuracy along with good parameter inference as shown in Table 2. ADF is extremely sensitive to good calibration for \( \sigma^2 \), without which estimates for \( \zeta \) are far from the truth and fail to converge by \( t = 500 \). Optimal performance is obtained by using the first data shard and performing the ADF approximation at \( t = 1 \) until convergence. These parameters estimates are subsequently propagated as described earlier. ADF produces accurate parameter point-estimates, however, severely underestimates uncertainty (see Table 2). This occurs in-part because ADF makes a global approximation on the joint posterior (restricting the propagation of uncertainty in a very specific way), whereas C-DF makes local approximations on the set of full conditionals.

<table>
<thead>
<tr>
<th></th>
<th>Avg. coverage ( \zeta )</th>
<th>Length</th>
<th>Time (sec)</th>
<th>( t = 200 )</th>
<th>( t = 400 )</th>
<th>( t = 500 )</th>
<th>MSE = ( \sum_{i=1}^{k} (\hat{\zeta}_i - \zeta_0)^2 / k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>0.87_{0.09}</td>
<td>0.53_{0.005}</td>
<td>85.0_{5.25}</td>
<td>0.77_{0.22}</td>
<td>0.29_{0.11}</td>
<td>0.26_{0.15}</td>
<td></td>
</tr>
<tr>
<td>SMCMC</td>
<td>0.92_{0.10}</td>
<td>0.52_{0.002}</td>
<td>119.4_{8.43}</td>
<td>0.41_{0.05}</td>
<td>0.22_{0.14}</td>
<td>0.20_{0.16}</td>
<td></td>
</tr>
<tr>
<td>ADF</td>
<td>0.36_{0.23}</td>
<td>0.80_{0.02}</td>
<td>0.88_{0.01}</td>
<td>0.42_{0.18}</td>
<td>0.28_{0.12}</td>
<td>0.27_{0.11}</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Inferential performance for C-DF, SMCMC, and ADF for parameter \( \zeta \). Coverage is based on 95% credible intervals averaged over all time points, all \( \zeta \) and over 10 independent replications. We report the time taken to produce 500 MCMC samples with the arrival of each data shard. MSE along with associated standard errors are reported at different time points.

### 3.2 Advanced data models

The C-DF algorithm introduced in Section 2 modifies the set of full conditional distributions to depend on propagated surrogate quantities which yields an approximate kernel. This
enables efficient online MCMC, with guarantees on correctness of approximate C-DF samples as data accrue over time (see Section 6). In settings where the conditional posterior is (a) not in closed form, or (b) corresponds to a model with an increasing number of parameters, theoretical consistency properties of parameters often do not hold. This section introduces extensions of the C-DF algorithm to three models of this type.

We present an application to binary regression in Section 3.2.2 where the conditional posterior distribution over the regression coefficients does not assume a closed form. For logistic regression, a variational approximation to the posterior introduces additional variational parameters for each observation to obtain a lower-bound for the likelihood (Jaakkola & Jordan, 2000). Additional non-conjugate message passing approximations have been considered in Braun and McAliffe (2010), and Tan (2014). One may also resort to ADF using
a Laplace approximation to the posterior over regression coefficients and propagating associated mean and covariance parameters in time. These techniques, however, are known to severely underestimate parameter uncertainty. Fortunately, data augmentation via the probit model enables conditionally conjugate sampling. Section 3.2.1 presents an application to a dynamic linear model for a continuous response centered on a first-order auto-regressive process. Here, the Kalman filtering provides an online sampling strategy. We discuss how to use the C-DF algorithm to overcome the computational and storage bottlenecks arising in both these examples. Additionally, Sections 3.2.1 and 3.2.2 provide excellent examples of C-DF applied to the conditionally conjugate and non-conjugate settings in increasing parameter spaces respectively. Finally, Section 3.2.3 considers an application to Poisson regression where no augmentation scheme is available and the full conditional distributions do not admit surrogate quantities. Here, we demonstrate how the online strategy of Luts et al. (2014) is a special case of the C-DF algorithm applied to a variational approximation of the posterior distribution.

3.2.1 Dynamic Linear Model

We consider the first-order dynamic linear model (West & Harrison, 1997), namely

\[
y_{t+1} \sim N(\theta_{t+1}, \sigma^2), \\
\theta_{t+1} \sim N(\phi \theta_t, \tau^2).
\]

Here, noisy observations \( y_t, t \geq 1 \) are observed and modeled as arising from an underlying stationary AR(1) process with lag parameter \( \phi, |\phi| < 1 \). Default priors \( \sigma^2 \sim IG(a_0, b_0), \tau^2 \sim IG(c_0, d_0), \phi \sim U(-1, 1) \) are chosen to complete the hierarchical model, and assume \( \theta_0 \sim N(0, h_0) \), the stationary distribution for the latent process. Forecasting future trajectories of the response is a common goal in this setting, hence good characterization of the distribution
over $\theta, \phi, \tau$ is of interest. Here, the full conditional distributions are given by

$$
\begin{align*}
\theta_{t+1}|y_{t+1}, \theta_t, \tau^2, \sigma^2, \phi &\sim N\left(\frac{\sigma^{-2} y_{t+1} + \tau^{-2} \phi \theta_t}{\sigma^{-2} + \tau^{-2}}, \frac{1}{\sigma^{-2} + \tau^{-2}}\right), \\
\theta_s|y_s, \theta_{s-1}, \theta_{s+1}, \tau^2, \sigma^2, \phi &\sim N\left(\frac{\sigma^{-2} y_{t+1} + \tau^{-2} \phi (\theta_{s-1} + \theta_{s+1})}{\sigma^{-2} + (\phi^2 + 1) \tau^{-2}}, \frac{1}{\sigma^{-2} + (\phi^2 + 1) \tau^{-2}}\right), 1 \leq s \leq t, \\
\sigma^2|\theta_{0:t}, y^t &\sim IG(a_t, b_t), \quad a_{t+1} = a_t + 1/2, \quad b_{t+1} = b_t + (y_{t+1} - \theta_{t+1})^2 / 2, \\
\tau^2|\theta_{0:t}, \phi &\sim IG(c_t, d_t), \quad c_{t+1} = c_t + 1/2, \quad d_{t+1} = d_t + (\theta_{t+1} - \phi \theta_t)^2 / 2, \\
\phi|\theta_{0:t}, \tau &\propto \pi(\phi) N\left(\frac{\sum_{s=1}^{t} \theta_s \theta_{s-1}}{\sum_{s=1}^{t} \theta_s^2}, \frac{\tau^2}{\sum_{s=1}^{t-1} \theta_s^2}\right) I(|\phi| < 1).
\end{align*}
$$

Forward-filtering and backward-sampling using Kalman filtering updates for the latent process enables online posterior computation. However, this strategy is computational intractable as the time horizon grows even for univariate time-series. While propagation of uncertainty for the latent process is of key importance, retrospective sampling of $\{\theta_s, s < t\}$ is often less meaningful. To extend the use of the C-DF algorithm in this growing parameter setting, we propose sampling of the latent process over a moving time-window of size $b$. We exploit the dependence structure of the latent process by borrowing strength across several observations to better estimate $\theta_t$. This eliminates the propagation of errors that might otherwise result from poor estimation at earlier time points. As the resampling window shifts forward, trailing latent parameters are fixed at their most recent point estimates. Parameter partitions for the C-DF algorithm must therefore also evolve dynamically, and are defined at time $t$ as $\Theta_{G_1} = \{\theta_t, \ldots, \theta_{t-b+1}, \tau^2, \sigma^2, \phi\}$, $\Theta_{G_2} = \{\theta_1, \ldots, \theta_{t-b}\}$, $t > b$. Unlike previous examples, conditional distribution $\pi(\phi|-)$ is not available in closed-form. Nevertheless, propagated surrogate quantities (SCSS) enable approximate MCMC draws to be sampled from C-DF full conditional $\tilde{\pi}(\phi|-)$ via Metropolis-Hastings. Here, steps for approximate MCMC using the C-DF algorithm in the context of a Metropolis-within-Gibbs sampler are:

1. At time $t$ observe $y_t$;

2. If $t \leq b$, draw $S$ Gibbs samples for $($$\theta_1, \ldots, \theta_t, \tau^2, \sigma^2, \phi$$)$ from the full conditionals. In
addition, \( C_1^t = C_2^t = C_3^t = C_4^t = 0, \ t \leq b. \)

(3) If \( t > b \), then for \( s = 1, \ldots, S \): (a) draw sequentially from \([\theta_{t-b+1} | \hat{\theta}_{t-b}, \theta_{t-b+2}, -], \ldots, [\theta_{t} | \theta_{t-1}, -]\);
(b) draw \( \tau^2 \sim IG(a_t, b_t) \), and \( b_t = C_4^{(t-1)} - 2\phi C_3^{(t-1)} + \phi^2 C_2^{(t-1)} + ((\theta_{t-b+1} - \phi \hat{\theta}_{t-b})^2 + \sum_{j=2}^{b}(\theta_{t+j-b} - \phi \theta_{t+j-b-1})^2)/2 \); (c) draw \( \sigma^2 \sim IG(a_t, C_4^{(t-1)} + \sum_{j=1}^{b}(y_{t+j-b} - \theta_{t+j-b})^2/2) \);
(d) sample \( \phi \sim \tilde{\pi}(\cdot | C_2^{(t-1)}, C_4^{(t-1)}, \theta_{(t-b+1):t}) \) via a Metropolis-Hastings step.

(4) Set \( \hat{\theta}_{t-b} \leftarrow \text{mean}(\theta_{t-b}) \);

(5) Update surrogate quantities \( C_1^t \leftarrow C_1^{(t-1)} + (y_{t-b} - \hat{\theta}_{t-b})^2/2, \ C_2^t \leftarrow C_2^{(t-1)} + \hat{\theta}_{t-b-1}/2, \ C_3^t \leftarrow C_3^{(t-1)} + \hat{\theta}_{t-b} \hat{\theta}_{t-b}, \ C_4^t = C_4^{(t-1)} + \hat{\theta}_{t-b}^2/2. \)

We vary signal-to-noise ratio \( \tau^2/\sigma^2 \) for the data generating process in simulation experiments. A number of high-signal cases were examined, and results are reported for a representative case with \( \tau = \sqrt{2}, \sigma = 0.1 \). A sufficiently large window size \( b \) is necessary to prevent C-DF from suffering due to poor estimation at early time-points, causing propagation of error in the defined surrogate quantities. For the PL competitor, 100 particles were propagated in time, and hence we fix \( b = 100 \) as well. Kernel density estimates for \( \theta_t \) using PL and the C-DF algorithm are shown in Figure 5. In the case of C-DF, estimates for all model parameters are found to be concentrated near their true values, whereas for PL, \( \theta_t \) at different times are centered correctly, albeit with much higher posterior variance (presumably due to poor estimation of noise parameters \( \tau^2, \sigma^2 \)). In addition to being 35% faster than PL, average coverage for the latent AR(1) process using the C-DF algorithm is near 80% with credible intervals roughly 10 times narrower than PL (see Table 3). C-DF is substantially more efficient in terms of memory and storage utilization, requiring only 6% of what PL uses for an identical simulation. Finally, C-DF produces accurate estimates for latent parameters \( \tau^2, \phi \) (in contrast to PL), although their posterior spread appears somewhat over shrunk. This may be remedied by using larger window size \( b \) at the expense of increased runtime which may be desired depending on the task at hand.
Table 3: Inferential performance for C-DF and Particle Learning. Coverage and length are based on 95% credible intervals for $\theta_t$ averaged over all time points and 10 independent replications. For truth $\theta_0$ at time $t$, we report $\text{MSE} = \frac{1}{T_n} \sum_{t=1}^{T_n} (\hat{\theta}_t - \theta_0)^2$. We report the time taken to run C-DF with 50 Gibbs samples at each time for $\tau^2, \theta, \sigma^2$ and 500 MH samples for $\phi$.

<table>
<thead>
<tr>
<th></th>
<th>Avg. coverage $\theta$</th>
<th>Length</th>
<th>Time (sec)</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>0.78_{0.10}</td>
<td>0.33_{0.11}</td>
<td>1138.60_{0.10}</td>
<td>0.011_{0.001}</td>
</tr>
<tr>
<td>PL</td>
<td>1_{0.00}</td>
<td>3.36_{0.46}</td>
<td>1750.58_{0.10}</td>
<td>0.096_{0.027}</td>
</tr>
</tbody>
</table>

Table 4: Computational and storage requirements for the Dynamic Linear Model using C-DF and PL. $C^t_{i,j}$, is the $i$-th CSS corresponding to the $j$-th particle in PL, $i = 1 : 4$, $j = 1 : N$, $N = 100$ is the number of particles propagated by PL, and $G = 500$ is the number of Metropolis samples used by both PL and C-DF. Memory in terms of RAM used to store and propagate SCSS and CSS for C-DF and PL is reported. Sampling and update complexities are in terms of big-O.

<table>
<thead>
<tr>
<th>Stats</th>
<th>Data</th>
<th>Sample complexity</th>
<th>Update complexity</th>
<th>Memory (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>$C^t_i$</td>
<td>${y_i}_{i&gt;nt-b}$</td>
<td>$S(N+G)$</td>
<td>128</td>
</tr>
<tr>
<td>PL</td>
<td>$C^t_{i,j}$</td>
<td>${y_i}_{i\geq 1}$</td>
<td>$NG$</td>
<td>3330</td>
</tr>
</tbody>
</table>

3.2.2 An application to binary response data

We consider an application of the C-DF algorithm for the probit model $\text{Pr}(y_i = 1|x_i) = \Phi(x'_i\beta)$, with standard normal distribution function $\Phi(\cdot)$. The Albert and Chib (1993) latent variable augmentation applies Gibbs sampling, with $y_i = \text{sign}(z_i)$, and $z_i|\beta \sim N(x'_i\beta, 1)$. Assuming a conditionally-conjugate prior, $\beta \sim N(0, \Sigma_\beta)$, the Gibbs sampler alternates between the following steps: (1) conditional on latent variables $z = (z_1, z_2, \ldots)'$, sample $\beta$ from its $p$-variate Gaussian full conditional; and (2) conditional on $\beta$ and the observed binary response $y$, impute latent variables from their truncated normal full conditionals. However, imputing latent variables $\{z_i : i \geq 1\}$ presents an increasing computational bottleneck as the sample size increases, as does recalculating the conditional sufficient statistics for $\beta$ given these latent variables at each iteration. C-DF alleviates this problem by imputing latent scores for the last $b$ training observations and propagating surrogate quantities. “Budget” size $b$ is allowed to grow with the predictor dimension, and as a default we set $b = p[\log(p)]$.

For data shards of size $n$, define $\mathcal{I}_t = \{i > nt - b\}$ as an index over the final $b$ observations.
Figure 3: Row #1 (left to right): Kernel density estimates for posterior draws of $\theta_t$ using PL and the C-DF algorithm at $t = 1000, 2000, 3000$; Row #2 (left to right) plots of model parameters $\tau^2$ and $\phi$, respectively.

at time $t$. Parameter partitions in this setting are dynamic as in Section 3.2.1, with $\Theta_{G_1} = \{\beta; z_i, i \in \mathcal{I}_t\}$ and $\Theta_{G_2} = \{z_i : i \leq nt - b\}$. The C-DF algorithm proceeds as follows:

1. Observe data $X^t, y^t$ at time $t$;

2. If $t = 1$, set $\beta = 0$, and draw $z_i \sim N(0, 1)$, $i = 1, \ldots, n$. If $t \leq b/n$, $C_t = 0$ and draw $S$ Gibbs samples for $(z_1, \ldots, z_{nt}, \beta)$ from the full conditionals.

3. If $t > b/n$, set $\beta \leftarrow \hat{\beta}_{t-1}$, update sufficient statistic $S_{XX}^t \leftarrow S_{XX}^{t-1} + X'^t X'^t$, and compute $\Sigma_{XX}^t = (S_{XX}^t + \Sigma^{-1})^{-1}$;

4. For $s = 1, \ldots, S$: (a) draw $z_i \sim TN(x_i'^t \beta)$, $i \in \mathcal{I}_t$, and define $z_{I_t}$ and $X_{I_t}$ respectively as the collection of latent draws and data for moving window $b$; (b) draw $\beta|z_{I_t}^{(s)}, C_{t-1} \sim N(\Sigma_{XX}^t C_t, \Sigma_{XX}^t)$, where $C_t(z_{I_t}^{(s)}) = C_{t-1} + X_{I_t}' z_{I_t}^{(s)}$;
(5) Set $\hat{\beta}_t \leftarrow \text{mean}(\beta^{(s)})$. For $u_i(\beta) = y_i\phi(-x'_i\beta)/\Phi(y_i,x'_i\beta)$, the trailing $n$ latent scores within moving window $b$ are fixed at their expected value, namely $\hat{z}_i \leftarrow x'_i\hat{\beta}_t + u_i(\hat{\beta}_t)$, $i \in T^\text{out}_t = \{(nt + b - 1) : (nt + 1) + b - 1\}$.

(6) Denote $X^\text{out}_t$ and $z^\text{out}_t$ as predictor data and latent scores for the “outgoing” set of data indexed by $T^\text{out}_t$. Update surrogate CSS $C_t \leftarrow C_{t-1} + X^\text{out}_t\hat{z}^\text{out}_t$, and set $\Theta_{\hat{G}_t} \leftarrow \Theta_{\hat{G}_t} \cup \hat{z}^\text{out}_t$.

We report simulation results for the following examples:

(Case 1) $(p, b, n, t) = (100, 500, 25, 100)$: $\beta_{j,0} \sim \text{Uniform}(-3/4, 3/4)$ for $j \in [11, 100]$;

(Case 2) $(p, b, n, t) = (500, 3500, 100, 100)$: $\beta_{j,0} \sim \text{Uniform}(-1/3, 1/3)$ for $j \in [11, 200]$.

The first 10 regression coefficients are $(3.5, -3.5, -2.0, 2.0, -1.5, 1.5, -1.5, 1.5, -1.0, 1.0)$, with $\beta_{j,0} = 0$ for $j > 200$ in case 2. Data are generated as $x_{ij} \sim N(0, \sigma = 0.25)$, and $P(y_i = 1) = \Phi(x_i\beta)$. Table 5 summarizes inferential performance for the regression parameters in each of the simulated cases. In case 2, although coverage for predictors with large coefficients (i.e., $\beta_1$ and $\beta_2$) is less than the nominal value, the average coverage across all predictors produced by C-DF is 70% despite the high dimension with a significant number of “noisy” predictors. In addition, C-DF has very good mean-square estimation of parameter coefficient in both cases. Figure 4 compares kernel density estimates of marginal posteriors at $t = 200, 500$ for both cases.

<table>
<thead>
<tr>
<th></th>
<th>Avg. coverage $\beta$</th>
<th>Length</th>
<th>Time (sec)</th>
<th>$\text{MSE}_{10}$</th>
<th>$\text{MSE}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Case 1 ($p = 100)$</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-DF</td>
<td>0.77$^{0.08}$</td>
<td>0.42</td>
<td>28.9$^{2.5}$</td>
<td>0.025$^{0.01}$</td>
<td>0.04$^{0.01}$</td>
</tr>
<tr>
<td>SMCMC</td>
<td>0.96$^{0.01}$</td>
<td>0.65</td>
<td>152.5$^{10.3}$</td>
<td>0.018$^{0.01}$</td>
<td>0.02$^{0.01}$</td>
</tr>
<tr>
<td><strong>Case 2 ($p = 500)$</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-DF</td>
<td>0.70$^{0.03}$</td>
<td>0.23</td>
<td>461.5$^{29.6}$</td>
<td>0.200$^{0.05}$</td>
<td>0.016$^{0.003}$</td>
</tr>
<tr>
<td>SMCMC</td>
<td>0.92$^{0.02}$</td>
<td>0.33</td>
<td>2196.3$^{170.5}$</td>
<td>0.050$^{0.01}$</td>
<td>0.009$^{0.001}$</td>
</tr>
</tbody>
</table>

Table 5: Inferential performance for C-DF and SMCMC in simulation studies $(p, b, n, t) = (100, 500, 25, 100)$ and $(p, b, n, t) = (500, 3500, 100, 100)$. Coverage and length are based on 95% credible intervals averaged over all predictors and over 10 independent replications. MSE is reported over all predictors, while $\text{MSE}_{10} = \frac{1}{10} \sum_{j=1}^{10} (\hat{\beta}_j - \beta_{j,0})^2$. We report the total time taken to produce 500 MCMC samples after the arrival of each data shard.
SMCMC scales linearly in the number of training observations and has the worst-case storage requirement. At each Gibbs iteration, draws for the latent scores is $O(nt)$ for SMCMC compared to $O(b)$ for C-DF. For both methods, sampling from the full conditional for $\beta$ is $O(p^3)$, updating sufficient statistics is $O(np^2)$, and updating surrogate quantities is $O(np)$. Computational and storage requirements for both methods are summarized in Table 6.

<table>
<thead>
<tr>
<th>Stats</th>
<th>Data</th>
<th>Sample complexity</th>
<th>Update complexity</th>
<th>Memory (M-bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>$S^{XX, C}$, $x_i, y_i_{i \geq nt-b}$</td>
<td>$Sb$</td>
<td>$p^3 + np^2$</td>
<td>20.0</td>
</tr>
<tr>
<td>SMCMC</td>
<td>$S^{XX}$, $x_i, y_i_{i \geq 1}$</td>
<td>$Snt$</td>
<td>$p^3 + np^2$</td>
<td>46.0</td>
</tr>
</tbody>
</table>

Table 6: Computational and storage requirements for the latent variable probit model using C-DF and SMCMC. Budget $b$ represents the number of latent scores updated by C-DF when processing data shard at time $t$. Runtime is quickly dominated by the sampling complexity which scales linearly in time for the augmented Gibbs sampler (SMCMC). Memory is reported for case 2, $(p, b, n, t) = (500, 3500, 100, 100)$. Sampling and update complexities are in terms of big-O.

Figure 4: Kernel density estimates for posterior draws using SMCMC and the C-DF algorithm at $t = 50, 100$. Shown from left to right are plots of model parameters $\beta_1$, $\beta_5$, and $\beta_{10}$ (and top to bottom are case 1 and case 2), respectively.
3.2.3 Poisson mixed effect model

Consider the Poisson additive mixed effects model where count data $y^t$ with predictors $X^t$ are modeled as

$$ y^t \sim \text{Poisson} \left( \exp(X^t \beta + Zu) \right), \quad \beta \sim N(0, \sigma_\beta^2 I_k) $$

$$ u \sim N(0, \text{diag}(\sigma_1^2 I_{k_1}, \ldots, \sigma_r^2 I_{k_r})), \quad \sigma_s^2 \sim IG(0.5, 1/b_s), \quad b_s \sim IG(1/2, 1/a_s^2), $$

with $I_{k_1}, \ldots, I_{k_r}$ respectively denoting identity matrices of order $k_1, \ldots, k_r$ and $s = 1, 2, \ldots, r$. Here, obstacles to implementing the C-DF algorithm immediately arise as the full conditional distributions have no closed-form and do not admit surrogate quantities to propagate. The global approximation made by the ADF approximation becomes increasingly unreliable and overly restrictive in higher dimensions. A possibly less restrictive approximation may seek to obtain an “optimal” approximation to the posterior subject to ignoring posterior dependence among different parameters. This is given by a variational approximation to the joint posterior over model parameters $\beta, u, \sigma_1^2, \ldots, \sigma_r^2$, namely

$$ \pi(\beta, u, \sigma_1^2, \ldots, \sigma_r^2 | D^t) \approx q_1(\beta, u) q_2(\sigma_1^2, \ldots, \sigma_r^2) q_3(b_1, \ldots, b_r). $$

Closed-form expressions for $q_1, q_2, q_3$ are derived and given by

$$ q_1 = N(\mu_{\beta,u}, \Sigma_{\beta,u}), \quad q_2 = \prod_{s=1}^r IG \left( \frac{k_s+1}{2}, \mu_{1/b_s} + \frac{||\mu_u||^2 + \text{tr}(\Sigma_u)}{2} \right), \quad q_3 = \prod_{s=1}^r IG \left( 1, \mu_{1/\sigma_s^2} + \sigma_s^{-2} \right). $$

Above, $\mu_{1/\sigma_s^2} = \int (1/\sigma_s^2) q(\sigma_s^2)$, with $\mu_{1/b_s}$ defined similarly, and $\mu_u, \Sigma_u$ represent the mean and covariance specific to $u$ under approximating density $q_1$. The approximate posterior is completely specified in terms of $\mu_{\beta,u}, \Sigma_{\beta,u}$, and $\mu_{1/b_s}, \mu_{1/\sigma_s^2}$ for $s = 1, \ldots, r$, hence the C-DF algorithm in this setup is applied directly on these parameters. In particular, one only needs point estimates for these parameters to fully specify the approximate posterior, hence sampling steps (see Algorithm 1) are replaced by fixed-point iteration. With a partition
\( \Theta_{g_1} = \{ \mu_{\beta,u}, \Sigma_{\beta,u} \}, \Theta_{g_2} = \{ \mu_{1/b_s}, \mu_{1/\sigma_{2}^2}, s = 1, \ldots, r \} \) for the parameters of the variational distribution, the C-DF algorithm proceeds as follows:

1. Observe data shard \((y^t, X^t)\) at time \(t\). If \(t = 1\), initialize all the parameters using draws from respective priors. Otherwise set \(\mu_{\beta,u}^{(t)} \leftarrow \mu_{\beta,u}^{(t-1)}, \Sigma_{\beta,u}^{(t)} \leftarrow \Sigma_{\beta,u}^{(t-1)}, \mu_{1/\sigma_{2}^2}^{(t)} \leftarrow \mu_{1/\sigma_{2}^2}^{(t-1)}, \mu_{1/b_s}^{(t)} \leftarrow \mu_{1/b_s}^{(t-1)}\);

2. The following steps are repeated a maximum of \(N_{fx}\) iterations (or until convergence):
   
   a. Update \(D \leftarrow \text{blockdiag}(\sigma_{\beta}^{-2}I_k, \mu_{1/\sigma_{2}^2}^t I_{t_1}, \ldots, \mu_{1/\sigma_{2}^2}^t I_{r_k})\);
   
   b. Update \(w_{\beta,u} \leftarrow \exp(c' \mu_{\beta,u} + \tfrac{1}{2} c' \Sigma_{\beta,u} c')\), with \(c' = [X^t, Z]\);
   
   c. Set \(H_1 = C_{1,1}^{(t-1)} + c' y^t, H_2 = C_{1,2}^{(t-1)} + c' w_{\beta,u}, \text{ and } H_3 = C_{1,3}^{(t-1)} + w_{\beta,u} c' c'\);
   
   d. Update \(\mu_{\beta,u} \leftarrow \mu_{\beta,u}^{(t-1)} + \Sigma_{\beta,u} (H_1 - H_2 - D \mu_{\beta,u}), \text{ and } \Sigma_{\beta,u} \leftarrow (H_3 + D)^{-1}\).

3. Set \(\mu_{\beta,u}^t, \Sigma_{\beta,u}^t\) at the final values after \(N_{fx}\) iterations;

4. Also repeat the following step a maximum of \(N_{fx}\) iterations (or until convergence):
   
   a. Update \(\mu_{1/b_s} \leftarrow (\mu_{1/\sigma_{2}^2} + a_s^{-2})^{-1}, \mu_{1/\sigma_{2}^2} \leftarrow (k_s + 1)/(2 \mu_{1/b_s} + \| \mu_{u}^{t} \|^2 + \text{tr}(\Sigma_{u}^t)), s = 1, \ldots, r\).

5. Set \(\mu_{1/\sigma_{2}^2}^t, \mu_{1/b_s}^t\) at the final values after \(N_{fx}\) iterations;

6. Update surrogate quantities \(C_{1,1}^t \leftarrow C_{1,1}^{(t-1)} + c' y^t, C_{1,2}^t \leftarrow C_{1,2}^{(t-1)} + c' w_{\beta,u}^t, C_{1,3}^t \leftarrow C_{1,3}^{(t-1)} + w_{\beta,u} c' c'\).

With the arrival of new data, \(N_{fx}\) fixed-point iterations update parameters estimates for the approximating distributions. Consistent with the definition of SCSS, an update to the parameters for \(\Theta_{g_1}\) under \(q_1\) may involve estimates from the previous time-point, in addition to estimates for parameters \(\Theta_{g_2}\) and vice versa. This online sampling scheme has excellent empirical performance, both in terms of parametric inference and prediction (see Luts & Wand (2014)). This establishes the broad applicability of propagating surrogate quantities.
(SCSS) using the C-DF algorithm in a non-conjugate setting using variational approximation methods.

4 C-DF for online compressed regression

We consider an application to compressed linear regression, where $y_1, \cdots, y_t \in \mathbb{R}^n$ are a sequence of $n$-dimensional response vectors with associated features $X_1, \cdots, X_t \in \mathbb{R}^{n \times p}$ observed over time. Data are modeled according to

$$y_t | \Phi, \beta, \sigma^2 \sim N(X_t \Phi' \beta, \sigma^2 I_n),$$

(7)

where $\Phi$ is an $m \times p$ projection matrix, with $m \ll p$. A Bayesian analysis proceeds by sampling from posterior $\beta, \Phi, \sigma^2 | D(t)$, with the following default prior specification: $\beta \sim N(0, \sigma^2 \Sigma_\beta)$; $\sigma^2 \sim IG(a, b)$, with Jeffrey’s prior obtained letting $a, b \to 0$; $\Phi \sim MN(\Phi_0, K, 1_m)$, centered on a row ortho-normalized random projection matrix, $\Phi_0$ with row-specific scaling $\kappa_i \overset{iid}{\sim} IG(1/2, 1/2)$ and $K = \text{diag}(\{\kappa_i\})$, $i = 1, 2, \ldots, m$. Of course, $\Phi$ and $\beta$ are only identified up to an orthogonal transformation, since $\Phi' \beta = \Phi'LL'\beta$ for any orthogonal matrix $L$. Nevertheless, regression coefficients $\gamma = \Phi' \beta$ are identifiable, and valid inference is obtained using posterior draws of the associated model parameters.

4.1 Online inference and competitors

Conditional on $\Phi$, the posterior distribution over $\beta, \sigma^2$ factorizes as

$$\beta | \Phi, D(t) \sim T_n(\mu_t, \Sigma_t) \quad \sigma^2 | \Phi, D(t) \sim IG(a_{1,t}/2, b_{1,t}/2)$$

$$\Sigma_t = b_{1,t}/n W^{-1} \quad a_{1,t} = nt$$

$$\mu_t = (b_{1,t}/n)^{-1} \Sigma_t \Phi F_t^{X'y'} \quad b_{1,t} = F_t^{yy} - F_t^{X'y} \Phi' W^{-1} \Phi F_t^{X'y'},$$

(8)
where $T_\nu(\cdot)$ is the multivariate t-distribution with $\nu$ degrees of freedom, with hyper-parameters defined in terms of sufficient statistics $F_{t}^{yy} = F_{t-1}^{yy} + y_t'y_t$, $F_{t}^{XY} = F_{t-1}^{XY} + y_t'X_t$, $F_{t}^{XX} = F_{t-1}^{XX} + X_t'X_t$ and $W = \Phi F_{t}^{XX} \Phi' + \Sigma^{-1}_\beta$. Sampling for $\Phi = [\Phi_1, \Phi_2, \ldots, \Phi_p]$ proceeds by drawing successively from the set of column-specific full conditionals, namely

$$
\Phi_j|\{\Phi\}_{-j}, K, \beta, D^{(t)} \sim N_m(\mu_{\Phi_j}, \Sigma_{\Phi_j}), \quad \kappa_i|\Phi \sim IG(c_i/2, d_i/2)
$$

$$
\Sigma_{\Phi_j} = \left(\sum_{s=1}^t \beta X_{js}'X_{js} \beta'/\sigma^2 + K^{-1}\right)^{-1}, \quad c_i = c + p
$$

$$
\mu_{\Phi_j} = \Sigma_{\Phi_j} \left(\sum_{s=1}^t \beta X_{js}'z_{js}/\sigma^2 + K^{-1}\Phi_{0j}\right), \quad d_i = d + (\Phi^{(i,:)} - \Phi_0^{(i,:)})'(\Phi^{(i,:)} - \Phi_0^{(i,:)}). \quad (9)
$$

For column $j$, $z_{jt} = y_t - \sum_{t\neq j} X_{lt} \Phi_{lt}/\beta$, with $X_{lt}$ denoting the $l$-th column of $X_t$, and $\Phi^{(i,:)}$ represents the $i$-th row of $\Phi$. The SMCMC sampling scheme for model (7) propagates sufficient statistics $F_{t}^{yy}, F_{t}^{XY}$ and $F_{t}^{XX}$ after observing $\{X_t, y_t\}$ at time $t$ and draws sequentially from full conditional distributions (8) and (9). Due to high autocorrelation between $\beta$ and $\Phi$, however, this online Gibbs sampler faces poor mixing in the joint parameter space. The C-DF algorithm applied to this setting partitions model parameters into $\Theta_{\varphi_1} = \{\beta, \sigma^2\}$ and $\Theta_{\varphi_2} = \{\Phi, K\}$, and inference proceeds as follows:

1. Observe data $D_t = \{X_t, y_t\}$ at time $t$. If $t = 1$, set $\beta = 0, \Phi = \Phi_0, \sigma = \bar{s}(\text{vec}(y_1, \ldots, y_k))$, and here we assume $y_t$ is zero-mean. Otherwise, set $\hat{\Phi}_t \leftarrow \hat{\Phi}_{t-1}, \hat{\kappa}_{i,t} \leftarrow \hat{\kappa}_{i,t-1}$;

2. Update surrogate quantities $C_{1,1}^{(t)} \leftarrow C_{1,1}^{(t-1)} + \hat{\Phi}_t X_t'X_t \hat{\Phi}_t$ and $C_{1,2}^{(t)} \leftarrow C_{1,2}^{(t-1)} + \Phi X_t'y_t$.

Using the notation in (8), redefine $W = C_{1,1}^{(t)} + \Sigma^{-1}_\beta$ and set $a_{1,t} = nt, b_{1,t} = F_{t}^{yy} - C_{1,2}^{(t)}W^{-1}C_{1,2}^{(t)'}. \quad (8)$

3. Draw $S$ approximate C-DF samples compositionally from $\sigma^2|\hat{\Phi}_t, C_{1,1}^{(t)}$ and $\beta|\sigma^2, \hat{\Phi}_t, C_{1,2}^{(t)}$.

Here, a single $m \times m$ matrix inversion is required (instead of once at every sample using SMCMC);

4. Set $\hat{\sigma}^2_t \leftarrow b_{1,t}/(a_{1,t}+1)$ and $\hat{\beta}_t \leftarrow \mu_t$, where $\Sigma_t = b_{1,t}/n W^{-1}$, and $\mu_t = (b_{1,t}/n)^{-1} \Sigma_t C_{1,2}^{(t)}$ (using closed-form expressions for the posterior MAP);
(5) For $j = 1, \ldots, p$: update surrogate quantities $C_{21,j}^{(t)} \leftarrow C_{21,j}^{(t-1)} + \hat{\beta}_t \hat{\beta}_t' (X_{jt}' X_{jt})$ and $C_{22,j}^{(t)} \leftarrow C_{22,j}^{(t-1)} + \hat{\beta}_t X_{jt}' y_t$.

(6) Draw $S$ samples from C-DF full conditional $\Phi, K | \hat{\beta}_t, \hat{\sigma}_t^2, C_2^{(t)}$ by sampling from $[\Phi_j | \{\Phi\}_{-j}, K, -]$, $1 \leq j \leq p$ and $[K | \Phi, -]$ defined in (9), now modified in terms the defined surrogate quantities:

(a) compute $H_{jt}(\Phi_{-j}) = C_{22,j}^{(t)} - \sum_{l \neq j} C_{21,l}^{(t)} \Phi_l$. Using (9), draw from $[\Phi_j | \{\Phi\}_{-j}, K, -]$ with $\Sigma_{\Phi,j} = (C_{21,j}^{(t)}/\hat{\sigma}_t^2 + K^{-1})^{-1}$ and $\mu_{\Phi,j} = \Sigma_{\Phi,j} (H_{jt}/\hat{\sigma}_t^2 + K^{-1} \Phi_0j)$;

(b) Sample $K | \Phi, -$ by drawing independently from $[\kappa_i | \Phi, -], i = 1, \ldots, m$.

(7) Set $\hat{\Phi}_t$ and $\{\hat{\kappa}_{i,t}\}$ as the sample mean over these $S$ draws.

By propagating surrogate quantities instead of the much larger sufficient statistics, the C-DF algorithm dramatically reduces storage requirements (see Section 4.2) and results in state-of-the-art parameter inference along with 20-40% gains in efficiency over SMCMC (see Table 11).

Bayesian shrinkage methods such as BL (Park et al., 2008) and GDP (Armagan et al., 2013) were attempted but rendered infeasible by the need to invert a $p \times p$ matrix at each MCMC iteration. SMC (Doucet et al., 2000) suffers from severe particle degeneracy for learning the high-dimensional parameter $\Phi$, and while sufficient statistics for particle rejuvenation are available, inference fairs no better than SMCMC. In addition, the need to store and propagate a large number of high-dimensional particles is completely impractical. Instead, we derive a variational Bayes (VB) approximation to the joint posterior using a GDP shrinkage prior on the coefficients of a standard linear regression model, $y_i \sim N(x_i^\prime \beta, \sigma^2)$ and $\beta_j \mid \sigma \sim GDP(\zeta = \frac{\sigma^2}{\alpha}, \alpha)$. The latter is equivalent to hierarchical prior $\beta_j \mid \sigma, \tau_j \sim N(0, \sigma^2 \tau_j)$, with $\tau_j \sim Exp(\lambda_j^2/2)$ and $\lambda_j \sim Ga(\alpha, \eta)$. For $\Theta = (\beta, \tau, \lambda, \sigma^2)'$, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_p)'$, and $\lambda = (\lambda_1, \ldots, \lambda_p)'$, we approximate $\pi(\Theta|D)$ by $q(\Theta) = \prod q_j(\theta_j)$. Given a variational approximation of this form, optimal densities $q_j(\theta_j) \propto \exp \left[ E_{-q(\theta_j)} \{ \log \pi(\Theta, D) \} \right]$ are obtained by
minimizing the KL distance between $\pi(\theta|D)$ and $q(\Theta)$. Here, $E_{q(\theta_j)}$ denotes the expectation over $\prod_{i \neq j} q_i(\theta_i)$. Additional details can be found in Ormerod & Wand (2010), Luts & Ormerod (2013).

4.2 Simulation experiments

The C-DF algorithm provides robust parameter inference and predictive performance across numerous simulation experiments which consider varying degrees of sparsity as well as predictor dimension and correlation (see Tables 8 and 10). Shards of $n = 100$ observations arrive sequentially over a $T = 1000$ time horizon, and predictor data are generated as $x_i \sim N(0,R)$, $R_{jk} = \rho^{|j-k|}$, for $j,k = 1,2,\ldots,p$ and correlation $\rho \in (0,1)$. The response is generated as $y_i \sim N(x_i'\beta,\sigma^2 = 4)$, where true coefficient vector $\beta$ used in each case is specified in Table 7.

<table>
<thead>
<tr>
<th>Case</th>
<th>$\rho$</th>
<th>$p$</th>
<th>$\beta_j \neq 0$</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1*</td>
<td>0.1</td>
<td>500</td>
<td>10</td>
<td>high</td>
</tr>
<tr>
<td>2*</td>
<td>0.1</td>
<td>1000</td>
<td>10</td>
<td>high</td>
</tr>
<tr>
<td>3*</td>
<td>0.4</td>
<td>500</td>
<td>10</td>
<td>high</td>
</tr>
<tr>
<td>4*</td>
<td>0.4</td>
<td>1000</td>
<td>10</td>
<td>high</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
<td>500</td>
<td>500</td>
<td>high</td>
</tr>
<tr>
<td>6</td>
<td>0.1</td>
<td>500</td>
<td>500</td>
<td>low</td>
</tr>
</tbody>
</table>

Table 7: Simulation experiments for supervised compressed regression. “High” signal strength corresponds to generating $\beta_j \sim \text{Uniform}(-3,3)$, while “low” signal corresponds to setting $\beta_j = 0.10$ for every nonzero feature. Above, sparse cases are denoted with (*).

Table 8 reports predictive MSE for each simulation experiment averaged over 50 simulated datasets. In all cases, the results appear robust to the choice of $m$, the dimension of the subspace which $\Phi$ maps into. Choosing $m$ large can add significantly to the computational overhead of the algorithm, so we suggest setting $m = \max\{10,\log(p)\}$ by default. VB-GDP yields the lowest MSPE for sparse truths, though C-DF and SMCMC demonstrate competitive performance. Increasing the number of predictors causes MSPE to increase for all methods, as does an increase of correlation between predictors. The C-DF algorithm
performs well in all cases, while VB-GDP suffers for dense truths, especially in the low-signal setting. In addition, C-DF results in excellent parameter estimation and variable selection, including cases with high predictor correlation (see Table 9). Table 10 reports coverage probabilities for 95% predictive intervals for the competing methods. While C-DF and SMCMC show proper coverage, VB suffers due to the restrictive assumption of independence between parameters \textit{a posteriori}.

<table>
<thead>
<tr>
<th></th>
<th>Case 1*</th>
<th>Case 2*</th>
<th>Case 3*</th>
<th>Case 4*</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>3.43,0.005</td>
<td>4.20,0.006</td>
<td>3.49,0.005</td>
<td>4.23,0.006</td>
<td>3.52,0.007</td>
<td>8.79,0.010</td>
</tr>
<tr>
<td>C-DF</td>
<td>3.49,0.010</td>
<td>4.38,0.010</td>
<td>3.62,0.007</td>
<td>4.40,0.020</td>
<td>3.81,0.006</td>
<td>3.64,0.020</td>
</tr>
<tr>
<td>SMCMC</td>
<td>3.56,0.020</td>
<td>4.40,0.020</td>
<td>3.58,0.020</td>
<td>4.43,0.020</td>
<td>3.68,0.020</td>
<td>3.70,0.020</td>
</tr>
</tbody>
</table>

Table 8: MSPE comparisons for each simulated experiment in Table 7. Subscripts denote bootstrapped standard errors calculated using independent replications.

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Case 1*</th>
<th>Case 2*</th>
<th>Case 3*</th>
<th>Case 4*</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>100</td>
<td>0.009,0.001</td>
<td>0.019,0.001</td>
<td>0.018,0.002</td>
<td>0.024,0.002</td>
<td>0.002,0.001</td>
<td>0.70,0.06</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.004,0.001</td>
<td>0.008,0.001</td>
<td>0.005,0.001</td>
<td>0.008,0.001</td>
<td>0.001,0.001</td>
<td>0.73,0.06</td>
</tr>
<tr>
<td>C-DF</td>
<td>100</td>
<td>0.010,0.001</td>
<td>0.033,0.004</td>
<td>0.029,0.004</td>
<td>0.074,0.001</td>
<td>0.013,0.001</td>
<td>0.042,0.003</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.004,0.001</td>
<td>0.011,0.002</td>
<td>0.011,0.002</td>
<td>0.027,0.001</td>
<td>0.003,0.001</td>
<td>0.020,0.001</td>
</tr>
<tr>
<td>SMCMC</td>
<td>100</td>
<td>0.014,0.003</td>
<td>0.032,0.004</td>
<td>0.024,0.003</td>
<td>0.034,0.009</td>
<td>0.004,0.001</td>
<td>0.063,0.006</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.006,0.001</td>
<td>0.015,0.002</td>
<td>0.010,0.002</td>
<td>0.017,0.004</td>
<td>0.002,0.001</td>
<td>0.036,0.002</td>
</tr>
</tbody>
</table>

Table 9: Performance comparison in terms of relative parameter MSE at \( t = 100, 200 \). Relative MSE for \( \gamma = \Phi \beta \) is computed as \( \| \hat{\beta}_{t} - \beta_0 \|^2 / \| \gamma_0 \|^2 \) for C-DF and SMCMC, \( \gamma_0 \) is \( \gamma \) at the truth. For VB, we compute \( \| \mu_{\beta} - \gamma_0 \|^2 / \| \gamma_0 \|^2 \), where \( \mu_{\beta} \) is the approximate posterior mean for \( \beta \). Subscripts denote bootstrapped standard errors calculated using independent replications.

<table>
<thead>
<tr>
<th></th>
<th>Case 1*</th>
<th>Case 2*</th>
<th>Case 3*</th>
<th>Case 4*</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>0.82</td>
<td>0.83</td>
<td>0.82</td>
<td>0.83</td>
<td>0.81</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>(0.81,0.84)</td>
<td>(0.76,0.87)</td>
<td>(0.81,0.85)</td>
<td>(0.75,0.87)</td>
<td>(0.76,0.86)</td>
<td>(0.80,0.85)</td>
</tr>
<tr>
<td>C-DF</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.97</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>(0.98,0.99)</td>
<td>(0.97,0.99)</td>
<td>(0.98,0.99)</td>
<td>(0.97,0.98)</td>
<td>(0.97,1)</td>
<td>(0.99,1)</td>
</tr>
<tr>
<td>SMCMC</td>
<td>0.96</td>
<td>0.93</td>
<td>0.95</td>
<td>0.93</td>
<td>0.96</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>(0.95,0.96)</td>
<td>(0.91,0.96)</td>
<td>(0.95,0.96)</td>
<td>(0.90,0.96)</td>
<td>(0.96,0.96)</td>
<td>(0.93,0.97)</td>
</tr>
</tbody>
</table>

Table 10: Predictive coverage for each simulation experiment in Table 7. Empirical 95% confidence intervals of the coverage probabilities over independent replications are also reported.
Propagating surrogate quantities using the C-DF algorithm for model (7) results in a dramatic efficiency gain over SMCMC. A measure of this efficiency is the “effective sample-size,” namely, the number of samples required for the predictive MSE to drop below a chosen threshold. Table 11 reports effective sample sizes for each simulated experiment in Table 7. Finally, storage of the sufficient statistics for model (7) in the predictor dimension, and updating these quantities is $O(np^2) + O(np)$ at each time point. In comparison, updating surrogate quantities for C-DF is $O(mpn) + O(m^2n)$. The C-DF algorithm dramatically reduces storage requirements for online inference by reducing the quadratic dependence on $p$ to a linear order.

<table>
<thead>
<tr>
<th></th>
<th>Case 1*</th>
<th>Case 2*</th>
<th>Case 3*</th>
<th>Case 4*</th>
<th>Case 5</th>
<th>Case 6</th>
<th>$p = 500$</th>
<th>$p = 1000$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>2040</td>
<td>2610</td>
<td>1710</td>
<td>3630</td>
<td>5250</td>
<td>210</td>
<td>0.68 MB</td>
<td>1.34 MB</td>
</tr>
<tr>
<td>SMCMC</td>
<td>2940</td>
<td>3990</td>
<td>2820</td>
<td>4410</td>
<td>4440</td>
<td>1050</td>
<td>2 MB</td>
<td>8.1 MB</td>
</tr>
</tbody>
</table>

Table 11: Effective sample sizes (number of MCMC samples) required until MSPE $\leq 5$ are shown above for each simulation experiment, with results averaged over 10 independent replications. The storage required in terms of propagated quantities is also reported for each competitor.

5 Real data illustrations

We consider two F16 aircraft control datasets having application to variable selection and prediction. For the “Ailerons” data, the goal is to predict the control action of the ailerons based on attributes related to the status of the airplane (e.g. climb-rate), while in the “Elevators” data, we predict the aircraft’s angle of elevation. Additional information is available at http://www.dcc.fc.up.pt/~ltorgo/Regression/DataSets.html. In both datasets, the relationship between the response and predictors is anticipated to be sparse. As before, VB-GDP and SMCMC are chosen as competitors and VB-GDP is run on the entire data in batch mode, while online algorithms feed in data shards sequentially over time. Approximate MCMC for model (7) obtained via the C-DF algorithm delivers good parameter inference.
and predictive performance as shown in Table 12. Effective sample size comparisons (not shown) between C-DF and SMCMC on both datasets are consistent with findings discussed in Section 4.2 and summarized in Table 11.

The Aileron dataset was partitioned with data shards having \( n = 50 \) observations arriving over a \( T = 143 \) time horizon. The predictor dimension is \( p = 40 \), and the hold-out test data has 6,595 observations. All methods are competitive in terms of predictive MSE, with online methods stabilizing within the first 10 time points. Predictive intervals have comparable coverage and length for C-DF and SMCMC, whereas the variational Bayes competitor VB-GDP once again suffers in this regard (see Table 12). Finally, VB-GDP results in a sparser solution (18 / 40 predictors selected), whereas 33 / 40 predictors have nonzero MAP estimates using C-DF and SMCMC.

For the Elevator data, 8750 training observations were partitioned with data shards having \( n = 175 \) observations with a predictor dimension of \( p = 14 \) arriving over a \( T = 50 \) time horizon. Again, C-DF and SMCMC have virtually identical performance in terms of MSPE, coverage and length of 95% PIs, however VB-GDP suffers in terms of predictive MSE and has predictive intervals which are more than twice as long than its competitors.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>MSPE</th>
<th>Coverage</th>
<th>PI length</th>
<th>#{\hat{\beta}_j = 0}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ailerons</td>
<td>VB</td>
<td>0.57</td>
<td>0.80</td>
<td>2.20</td>
<td>22/40</td>
</tr>
<tr>
<td></td>
<td>C-DF</td>
<td>0.58</td>
<td>0.93</td>
<td>2.86</td>
<td>7/40</td>
</tr>
<tr>
<td></td>
<td>SMCMC</td>
<td>0.57</td>
<td>0.94</td>
<td>2.89</td>
<td>7/40</td>
</tr>
<tr>
<td>Elevators</td>
<td>VB</td>
<td>0.94</td>
<td>0.98</td>
<td>9.68</td>
<td>4/14</td>
</tr>
<tr>
<td></td>
<td>C-DF</td>
<td>0.89</td>
<td>0.96</td>
<td>3.74</td>
<td>1/14</td>
</tr>
<tr>
<td></td>
<td>SMCMC</td>
<td>0.90</td>
<td>0.96</td>
<td>3.73</td>
<td>2/14</td>
</tr>
</tbody>
</table>

Table 12: Predictive performance for real data illustrations after training on the full data. As before, MSPE denotes the mean square prediction error measured on a hold-out dataset. Coverage and length are based on 95% predictive intervals averaged over all time points on a hold-out dataset. \#\{\hat{\beta}_j = 0\} refers to the number of features identified by the model as unrelated to the response. For Ailerons (Elevators) data, the hold-out dataset consists of 6597 (7896) observations.
6 Convergence Behavior of Approximate Samplers

This section studies the convergence behavior for a general class of approximate MCMC algorithms, with the C-DF algorithm a special case. We first present a general result on the limiting error of approximation when a kernel with the targeted stationary distribution is approximated by another kernel. This result deals with finite samples, and is not limited to any specific class of approximations. We then show that such approximations improve with increasing sample size, yielding draws from the exact posterior asymptotically. Proofs are provided in the appendix.

6.1 Notation and Framework

Let $π_t(·|D^{(t)})$ be posterior distribution having observed data $D^{(t)}$ through time $t$. A sequence of probability measures $π_t(·|D^{(t)})$ is defined on a corresponding sequence of measure spaces, $(H_t, H_t)$, $t ≥ 1$. Below, we take $H_t = R^p$, with $H_t = B(R^p)$ denoting the Borel $σ$-algebra on $R^p$ for a fixed $p$-dimensional parameter space $Θ = (θ_1, · · · , θ_p)$. $π_t$ admits density $π_t(Θ)$ with respect to the Lebesgue measure $dν(Θ) = dν_1(θ_1), d(θ_2), · · · , dν_p(θ_p)$. Transition kernel $T_t : R^p × R^p → R^+$ defines the parameter updating process at time $t$, and

1. $T_t(x, ·)$ is a probability measure for all $x ∈ R^p$;

2. $T_t(·, A)$ is a measurable function w.r.t the $σ$-algebra for all $ν$-measurable $A$.

A function $f_t : R^p → R^+$ defined over $σ$-field $H_t$ is the invariant distribution of $T_t$ if

$$f_t(Θ') = \int T_t(Θ, Θ')f_t(Θ)dν(Θ).$$

(10)

To denote the transition kernels and stationary distributions for finite sample cases, we omit the subscript $t$. We study the convergence of a sequence of distributions in total variation norm, namely $d_{TV}(μ_1, μ_2) = \sup_A |μ_1(A) − μ_2(A)|$. 

31
6.2 A finite sample error bound for approximate MCMC samplers

We begin by characterizing how error propagates while approximating a kernel by another kernel.

**Lemma 6.1** Let $K$ be a kernel approximated by a kernel $T$ s.t.

$$\sup_{\Theta} ||K(\Theta, \cdot) - T(\Theta, \cdot)||_{TV} \leq \rho.$$ 

Assume, $\mu_1, \mu_2$ are the stationary distributions of $T$ and $K$ respectively. Further assume, $||T(r) - \mu_1||_{TV} \to 0$ and $||K(r) - \mu_2||_{TV} \to 0$. Then $\exists r_0$ s.t.

$$\sup_{\Theta} ||K^{(r)}(\Theta, \cdot) - T^{(r)}(\Theta, \cdot)||_{TV} \leq r\rho, \forall r \leq r_0.$$

$$\sup_{\Theta} ||K^{(r)}(\Theta, \cdot) - T^{(r)}(\Theta, \cdot)||_{TV} \leq \min\{\rho r, 2||\mu_1 - \mu_2||_{TV}\}, \forall r > r_0.$$ 

(11)

When the sample size is small, with data observed over a finite time horizon or at once, the C-DF kernel remains unchanged once surrogate quantities (SCSS) have been calculated for the final shard of data. According to Lemma 6.1, the approximation error due to C-DF increases initially before stabilizing. With certain additional assumptions, the C-DF kernel can be shown to generate draws from the exact posterior distribution as data accumulates over time (i.e., for streaming data).

6.3 Convergence results for a general approximation class

We assume a $p$-dimensional parameter $\Theta$ is partitioned into $\Theta_1 = (\theta_{11}, \ldots, \theta_{1p_1})' \in \mathbb{R}^{p_1}$ and $\Theta_2 = (\theta_{21}, \ldots, \theta_{2p_2})' \in \mathbb{R}^{p_2}$, where $p = p_1 + p_2$. The C-DF approximation requires that the approximating kernel for a full conditional distribution admits surrogate quantities as discussed in Section 2. Within this framework and without loss of generality, we assume that $\theta_{1i} | \theta_{1,-i}, \hat{\Theta}_2, i = 1, \ldots, p_1$ are conditionally conjugate distributions. Then, at time $t$ the approximating kernel $T_t : \mathbb{R}^{p_1} \times \mathbb{R}^{p_2} \to \mathbb{R}^+$ can be of the following forms:
(1) All full conditional distributions $\theta_{2i} | \theta_{2,-i}, \hat{\Theta}_1, i = 1, \ldots, p_2$ are conjugate, hence sampling approximate transition kernel $T$ proceeds in a Gibbs-like fashion. For two sequences of estimators $\{\hat{\Theta}_{1,t}\}_{t=1}^{\infty}$, $\{\hat{\Theta}_{2,t}\}_{t=1}^{\infty}$, the C-DF transition kernel $T_t$ is defined as

$$T_t(\Theta, \Theta') = \prod_{i=1}^{p_1} \pi_t(\theta'_{1i} | \hat{\Theta}_{2,t-1}, \theta'_{1l}, l < i, \theta_{1l}, l > i) \prod_{i=1}^{p_2} \pi_t(\theta'_{2i} | \hat{\Theta}_{1,t-1}, \theta'_{2l}, l < i, \theta_{2l}, l > i).$$

(12)

(2) Some (or all) of the conditional distributions $\theta_{2i} | \theta_{2,-i}, \hat{\Theta}_1, i = 1, \ldots, p_2$ are non-conjugate (i.e., not available in closed-form). The C-DF transition kernel $T_t$ is defined as

$$T_t(\Theta, \Theta') = \prod_{i=1}^{p_1} \pi_t(\theta'_{1i} | \hat{\Theta}_{2,t-1}, \theta'_{1l}, l < i, \theta_{1l}, l > i) \prod_{i=1}^{p_2} \pi_t(\theta'_{2i} | \hat{\Theta}_{1,t-1}, \theta'_{2l}, l < i, \theta_{2l}, l > i) Q(\Theta_2, \Theta'_2 | \hat{\Theta}_{1,t-1}).$$

(13)

Here, $\Theta_2$ is updated using a Metropolis-Hastings step having kernel $Q(\Theta_2, \Theta'_2 | \hat{\Theta}_1)$. As before, $\{\hat{\Theta}_{1,t}\}_{t=1}^{\infty}$, $\{\hat{\Theta}_{2,t}\}_{t=1}^{\infty}$ are two sequences of estimators.

Lemma (6.2) specifies the unique stationary distribution $f_t : \mathbb{R}^p \rightarrow \mathbb{R}^+$ of $T_t$.

Lemma 6.2: C-DF approximate kernel $T_t$ in (12) and (13) have unique stationary distribution $f_t(\Theta) = \pi_t(\Theta | \hat{\Theta}_{2,t-1}) \pi_t(\Theta_2 | \hat{\Theta}_{1,t-1})$, where $\pi_t(\Theta | -)$ and $\pi_t(\Theta_2 | -)$ are the stationary distributions for the respective parameter full conditionals.

Remark 1: Illustrations in Section 3.1 fall into the first scenario, where all model parameters have conjugate full conditionals that admits surrogate quantities (SCSS). The dynamic linear model presented in Section 3.2.1 is an example where full conditional distributions admit SCSS, although some require sampling via Metropolis-Hastings. For all other examples presented in Section 3.2, the C-DF algorithm is modified to accommodate situations in which (a) the parameter space is increasing over time; or (b) some (or all) of the full conditional distributions do not admit SCSS. Though asymptotic guarantees on the samples produced are not established by Theorem (6.3) for the latter cases, the C-DF algorithm

33
proves its versatility by producing excellent inferential and predictive performance in all examples considered (see Section 3).

Let $\pi_0$ be the initial distribution from which parameters are drawn. Below, we state the main result.

**Theorem 6.3** Assume, (i) $\exists \alpha_t \in (0, 1)$ s.t $\forall t, \sup_{\Theta} d_{TV}(T_t(\Theta, \cdot), f_t) \leq 2\alpha_t$, (ii) $d_{TV}(f_t, f_{t-1}) \to 0$, and (iii) $d_{TV}(f_t, \pi_t) \to 0$, then $\exists \{n_t\}_{t \geq 1}$ s.t. $d_{TV}(T^{(n_t)}_t \cdots T^{(n_1)}_1, \pi_0, \pi_t) \to 0$.

**Remark 2:** In essence, Theorem 6.3 states that running a Markov chain with approximate kernel $T_t$ for $n_t$ times at each time $t$ will asymptotically have draws from the true joint posterior distribution.

**Remark 3:** Condition (i) in Theorem 6.3 is referred to as the universal ergodicity condition (Yang & Dunson, 2013). It is shown in Lemma 3.2 in Yang & Dunson (2013) that the universal ergodicity condition is weaker than uniform ergodicity condition on the transition kernel $T$. Condition (ii) ensures that the stationary distribution of the approximating kernel changes slowly as time progresses. Lemma 3.7 in Yang & Dunson (2013) shows that condition (ii) is satisfied for any regular parametric model by applying a simple Bernstein-Von Mises theorem. Finally, the stationary distribution of the approximating chain is required to be “close” to the true posterior distribution at later time; this is formalized in condition (iii). Sufficient conditions under which assumption (iii) holds are outlined in Lemma 6.4. Before stating this Lemma, we recall the definition of posterior consistency.

**Definition:** A posterior $\Pi(\cdot | D^{(t)})$ is defined to be consistent at $\Theta^0$ if, for every neighborhood $U$ of $\Theta^0$, $\Pi(U | D^t) \to 1$ under the data generating law at $\Theta^0$.

**Lemma 6.4** Assume that the likelihood function $p_{\Theta}(\cdot)$ is continuous as a function of $\Theta$ at $\Theta^0 = (\Theta_1^0, \Theta_2^0)$ and $\sqrt{t}p_{\Theta}(D^{(t)})$ in limit is bounded away from 0 and $\infty$. Suppose $\Theta^0$ is an interior point in the domain, with the prior distribution $\pi_0(\Theta_1, \Theta_2)$ being positive and...
continuous at $\Theta^0$. Further, assume $\hat{\Theta}_{1,t} \rightarrow \Theta^0_1$, $\hat{\Theta}_{2,t} \rightarrow \Theta^0_2$ a.s. under the data generating law at $\Theta_0$, and $f_t$ and $\pi_t$ are both consistent at $\Theta^0$. Then

$$\int |\pi_t(\Theta) - f_t(\Theta)| \, d\Theta \rightarrow 0 \text{ as } t \rightarrow \infty,$$

almost surely under the true data generating model at $\Theta^0$.

**Remark 4:** Using the simple fact that

$$
    d_{TV}(\pi_t, f_t) = 2 \sup_A |\int_A (\pi_t - f_t)| = 2 \int_{\pi_t > f_t} (\pi_t - f_t) = \int_{\pi_t > f_t} (\pi_t - f_t) + \int_{f_t > \pi_t} (f_t - \pi_t) \\
    = \int |\pi_t - f_t|,
$$

it is clear that under the conditions of lemma 6.4, $d_{TV}(\pi_t, f_t) \rightarrow 0$ as $t \rightarrow \infty$.

**Remark 5:** Lemma 6.4 says that if the likelihood grows at a certain rate (satisfied under standard regularity conditions) and the estimates $\hat{\Theta}_{1,t}, \hat{\Theta}_{2,t}$ are consistent estimators of true parameters, the stationary distribution of C-DF becomes close to the stationary distribution of Gibbs sampling on shards. This means that the condition (iii) in Theorem 6.3 is getting satisfied. It should also be that our primary focus is on static models where it is quite easy to obtain consistent sequence of estimates for the involved parameters (as shown in examples).

## 7 Conclusion

Conditional density filtering (C-DF) is a novel algorithm designed to scale Bayesian inference to streaming data. C-DF produces approximate MCMC samples by drawing from approximations to the conditional posterior distributions which rely on propagated surrogate conditional sufficient statistics (SCSS). These quantities are computed using sequential point estimates for model parameters along with data shards observed in time. This eliminates
the need to process the entire data set simultaneously, with potentially substantial computational gains for massive data sets that are distributed across a computer network. To date, there have been a limited number of approaches that scale Bayesian inference to data with a very large number of observations. Popular approaches that often work well in simple models include assumed density filtering (ADF), variational Bayes, expectation propagation (EP), integrated nested Laplace approximation (INLA), along with particle filtering and sequential Monte Carlo (SMC). These approaches face substantial difficulty, either computationally or in terms of inferential accuracy, for more complex models (i.e., non-conjugate settings or when the parameter space becomes large) and come with no asymptotic guarantee (with the exception of SMC). In contrast, approximate MCMC using C-DF is shown to have the correct stationary distribution as data accrue over time (see Section 6). The versatility of C-DF algorithm is demonstrated through illustrative examples presented in Sections 3.1, 3.2 and 4.1. In each, approximate sampling via the C-DF algorithm results in good parameter estimation, uncertainty characterization, and predictive performance. In addition, C-DF offers runtime, memory and sampling efficiency improvements compared to various other competitors (see supporting Tables in Sections 3.2.1, 3.2.2, and 4.1).
Appendix

A  Proofs

Proof of Lemma 6.1

The proof follows by induction. First we will prove an identity that will be used in the proof that follows. Letting $A \in B(\mathcal{R}^d)$,

$$
K^{(r)}(\Theta, A) - T^{(r)}(\Theta, A) = \int \left[ K^{(r-1)}(\Theta', A) - T^{(r-1)}(\Theta', A) \right] T(\Theta, d\Theta') + \int \left[ K(\Theta, d\Theta') - T(\Theta, d\Theta') \right] K^{(r-1)}(\Theta', A).
$$

(14)

(14) relates the differences between the kernels at $r$-th iteration of the Markov chain. Using the fact that r.h.s is free of $A$ and the relationship that $||\nu_1 - \nu_2||_{TV} = \sup_{g: \mathcal{R}^d \to [0,1]} |\int g d\nu_1 - \int g d\nu_2|$, (14) yields

$$
||K^{(r)}(\Theta, \cdot) - T^{(r)}(\Theta, \cdot)||_{TV} \leq ||K(\Theta, \cdot) - T(\Theta, \cdot)||_{TV} + \int ||K^{(r-1)}(\Theta', \cdot) - T^{(r-1)}(\Theta', \cdot)||_{TV} T(\Theta, d\Theta').
$$

(15)

Suppose (11) holds for $(r-1)$. Using (15) we find

$$
\sup_{\Theta} ||K^{(r)}(\Theta, \cdot) - T^{(r)}(\Theta, \cdot)||_{TV} \leq \sup_{\Theta} ||K(\Theta, \cdot) - T(\Theta, \cdot)||_{TV}
$$

$$
+ \sup_{\Theta} \int ||K^{(r-1)}(\Theta', \cdot) - T^{(r-1)}(\Theta', \cdot)||_{TV} T(\Theta, d\Theta')
$$

$$
\leq \sup_{\Theta} ||K(\Theta, \cdot) - T(\Theta, \cdot)||_{TV} + (r-1) \rho < \rho + (r-1) \rho = \rho r.
$$

(16)
Also note that $\exists r_0$ s.t. for all $r > r_0$, $||T^{(r)} - \mu_1||_{TV} < \frac{1}{2}||\mu_1 - \mu_2||_{TV}$ and $||K^{(r)} - \mu_2||_{TV} < \frac{1}{2}||\mu_1 - \mu_2||_{TV}$. Using triangle inequality, for all $r > r_0$

$$||K^{(r)} - T^{(r)}||_{TV} \leq ||K^{(r)} - \mu_2||_{TV} + ||\mu_1 - \mu_2||_{TV} + ||T^{(r)} - \mu_1||_{TV} < 2||\mu_1 - \mu_2||_{TV}. \quad (17)$$

Comparing (16) and (17) the result follows.

**Proof of Lemma (6.2)**

We will show that $\int T(t(\Theta, \Theta')) f_t(\Theta) d\Theta = f_t(\Theta')$. Note that in case (i)

$$L.H.S = \int \prod_{i=1}^{d_1} \pi_t(\Theta'_{1i} | \Theta_{2,t-1}, \Theta'_{1l}, l < i, \Theta_{1l}, l > i) \left[ \prod_{i=1}^{d_2} \pi_t(\Theta'_{2i} | \Theta_{1,t-1}, \Theta'_{2l}, l < i, \Theta_{2l}, l > i) \right] \pi_t(\Theta_{1,t-1} | \Theta_{2,t-1}) \pi_t(\Theta_{2} | \Theta_{1,t-1})$$

$$= \left( \int \prod_{i=1}^{d_1} \pi_t(\Theta'_{1i} | \Theta_{2,t-1}, \Theta'_{1l}, l < i, \Theta_{1l}, l > i) \right) \pi(\Theta_{1,t-1} | \Theta_{1,t-1}) d\Theta_1$$

$$= \pi_t(\Theta'_{1i} | \Theta_{2,t-1}) \pi_t(\Theta_{2} | \Theta_{1,t-1}).$$

The last step follows from the well known fact that $\left[ \prod_{i=1}^{d_2} \pi_t(\Theta'_{2i} | \Theta_{1,t-1}, \Theta'_{2l}, l < i, \Theta_{2l}, l > i) \right]$, $\left[ \prod_{i=1}^{d_1} \pi_t(\Theta'_{1i} | \Theta_{1,t-1}, \Theta'_{1l}, l < i, \Theta_{1l}, l > i) \right]$ are the Gibbs sampling kernels with the stationary distribution $\pi_t(\Theta_{2} | \Theta_{1,t-1})$ and $\pi_t(\Theta_{1} | \Theta_{2,t-1})$ respectively. The proof for case (ii) follows in an identical manner taking into account that the MH kernel $Q(\Theta_{2}', \Theta_{2} | \Theta_{1,t-1})$ has $\pi_t(\Theta_{2} | \Theta_{1,t-1})$ as its stationary distribution.

**Proof of Lemma (6.3)**

The following proof builds on results from Yang & Dunson (2013). Although most of the proof coincides with the proof of Theorem 3.6 in Yang & Dunson (2013), for the sake of completeness we present the entire proof. Fix $\epsilon \in (0, 1)$. Choose $n_t$, $t \geq 1$ s.t. $\rho^{n_t} < \epsilon$. Using
the fact that universal ergodicity condition implies uniform ergodicity, one obtains
\[ d_{TV}(T_t^{n_t}, \pi_t) \leq a_t^{n_t} < \epsilon. \]

Let \( h = T_{t-1}^{n_{t-1}} \cdots T_1^{n_1} \pi_0 \), then

\[ d_{TV}(T_t^{n_t} \cdots T_1^{n_1} \pi_0, f_t) = d_{TV}(T_t^{n_t} h, f_t) \leq d_{TV}(T_t^{n_t}, f_t) d_{TV}(h, f_t) \leq a_t^{n_t} d_{TV}(h, f_t) \leq \epsilon (d_{TV}(h, f_{t-1}) + d_{TV}(f_t, f_{t-1})). \quad (18) \]

using the result repeatedly, one obtains

\[ d_{TV}(T_t^{n_t} \cdots T_1^{n_1} \pi_0, f_t) \leq \epsilon^{t+1-t} d_{TV}(f_t, f_{t-1}). \]

R.H.S clearly converges to 0 applying condition (ii). The proof is completed by using condition (iii) and the fact that

\[ d_{TV}(T_t^{n_t} \cdots T_1^{n_1} \pi_0, \pi_t) \leq d_{TV}(T_t^{n_t} \cdots T_1^{n_1} \pi_0, f_t) + d_{TV}(f_t, \pi_t). \]

**Proof of Lemma (6.4)**

Note that the approximated posterior distribution \( f_t \) is given by

\[
\begin{align*}
    f_t(\Theta_1, \Theta_2) &= \pi_t(\Theta_1 | \hat{\Theta}_{2,t}) \pi_t(\Theta_2 | \hat{\Theta}_{1,t}) \\
    &= \left[ \prod_{l=1}^{t} p_{\Theta_1, \hat{\Theta}_{2,t}}(D_l) \right] \left[ \prod_{l=1}^{t} p_{\Theta_1, \hat{\Theta}_{2,t}}(D_l) \right] \left[ \prod_{l=1}^{t} p_{\Theta_1, \hat{\Theta}_{2,t}}(D_l) \right]\pi_0(\hat{\Theta}_{1,t}, \Theta_2) \pi_0(\Theta_1, \hat{\Theta}_{2,t}) \\
    &\int \left[ \prod_{l=1}^{t} p_{\Theta_1, \hat{\Theta}_{2,t}}(D_l) \right] \left[ \prod_{l=1}^{t} p_{\Theta_1, \hat{\Theta}_{2,t}}(D_l) \right]\pi_0(\hat{\Theta}_{1,t}, \Theta_2) \pi_0(\Theta_1, \hat{\Theta}_{2,t}).
\end{align*}
\]

Given that \( \hat{\Theta}_{1,t} \rightarrow \Theta_1^0, \hat{\Theta}_{2,t} \rightarrow \Theta_2^0 \) a.s. under \( \Theta^0, \) \( \exists \Omega_0 \) which has prob. 1 under the data generating law s.t. \( \forall \omega \in \Theta^0, \hat{\Theta}_{1,t}(\omega) \) and \( \hat{\Theta}_{2,t}(\omega) \) are in the arbitrary small neighborhood of \( \Theta_1^0 \) and \( \Theta_2^0 \) respectively.

As \( \pi_0 \) is cont. at \( \Theta^0, \) given \( \epsilon > 0 \) and \( \eta > 0 \) \( \exists \) a neighborhood \( N_{\epsilon, \eta} \) s.t. \( \forall \Theta \in N_{\epsilon, \eta} \) one
has

\[ |\pi_0(\Theta_1, \Theta_2) - \pi_0(\Theta_1^0, \Theta_2^0)| < \epsilon. \]  \hspace{1cm} (19)

Using (19) and consistency of \( \hat{\Theta}_{1,t} \) and \( \hat{\Theta}_{2,t} \) one obtains, for all \( t > t_0 \) and \( \omega \in \Omega_0 \)

\[ |\pi_0(\Theta_1, \hat{\Theta}_{2,t}) - \pi_0(\Theta_1^0)| < \epsilon, \]  \hspace{1cm} (20)

Continuity of \( p_\Theta(\cdot) \) at \( \Theta^0 \) leads to the condition that for all \( t > t_0, \)

\[ |p_{\Theta_1, \Theta_2}(D_t) - p_{\Theta_1^0, \Theta_2^0}(D_t)| < \epsilon. \]  \hspace{1cm} (21)

Also consistency assumptions on \( f_t \) and \( \pi_t \) yield that for all \( t > t_1 \) and \( \omega \in \Omega_1 \)

\[ f_t(N_{\epsilon, \eta}|D_t(\omega)) > 1 - \eta, \]  \hspace{1cm} \pi_t(N_{\epsilon, \eta}|D_t(\omega)) > 1 - \eta,

where \( \Omega_1 \) has probability 1 under the data generating law. Considering \( \Omega = \Omega_0 \cap \Omega_1 \) and \( t_2 = \max\{t_1, t_0\} \) it is evident that \( \Omega \) has also prob. 1 under the true data generating law and all of the above conditions hold for \( t > t_2 \) and \( \omega \in \Omega \).

Simple algebraic manipulations yield

\[
\frac{f_t(\Theta|D_t(\omega))}{\pi_t(\Theta|D_t(\omega))} = \frac{f_t(N_{\epsilon, \eta}|D_t(\omega))}{\pi_t(N_{\epsilon, \eta}|D_t(\omega))} \times \left[ \frac{\prod_{l=1}^t p_{\Theta_1, \Theta_2,t}(D_l)}{\int_{N_{\epsilon, \eta}} \prod_{l=1}^t p_{\Theta_1, \Theta_2,t}(D_l)} \right] \frac{\pi_0(\hat{\Theta}_{1,t}, \Theta_2) \pi_0(\Theta_1, \hat{\Theta}_{2,t})}{\pi_0(\hat{\Theta}_{1,t}, \Theta_2) \pi_0(\Theta_1, \hat{\Theta}_{2,t})} \left[ \frac{\int_{N_{\epsilon, \eta}} \prod_{l=1}^t p_{\Theta}(D_l) \pi_0(\Theta)}{\int_{1} \prod_{l=1}^t p_{\Theta}(D_l) \pi_0(\Theta)} \right]
\]

\hspace{1cm} (22)
Using (20) we have

\[
(\pi_0(\Theta^0) - \epsilon)^2 \int \prod_{l=1}^t p_{\Theta_1,\Theta_2,t}(D_l)p_{\Theta_1,t,\Theta_2}(D_l) \\
\leq \int \prod_{l=1}^t \left[ p_{\Theta_1,\Theta_2,t}(D_l)p_{\Theta_1,t,\Theta_2}(D_l) \right] \pi_0(\Theta_1,t,\Theta_2)\pi_0(\Theta_1,\Theta_2,t) \\
\leq (\pi_0(\Theta^0) + \epsilon)^2 \int \prod_{l=1}^t p_{\Theta_1,\Theta_2,t}(D_l)p_{\Theta_1,t,\Theta_2}(D_l).
\]

Similarly,

\[
(\pi_0(\Theta^0) - \epsilon) \int \prod_{l=1}^t p_\Theta(D_l) \leq \int \prod_{l=1}^t \left[ p_\Theta(D_l) \right] \pi_0(\Theta) \leq (\pi_0(\Theta^0) + \epsilon) \int \prod_{l=1}^t p_\Theta(D_l).
\]

Therefore,

\[
\frac{f_t(\Theta|D^{(t)}(\omega))}{\pi_t(\Theta|D^{(t)}(\omega))} \leq (1 - \eta)^{-1} \left[ \frac{\prod_{l=1}^t p_{\Theta_1,\Theta_2,t}(D_l)p_{\Theta_1,t,\Theta_2}(D_l)}{\int \prod_{l=1}^t p_\Theta(D_l)} \right] \left[ \frac{\int \prod_{l=1}^t p_\Theta(D_l)}{\prod_{l=1}^t p_\Theta(D_l)} \right] \frac{\left(\pi_0(\Theta^0) + \epsilon\right)^3}{\left(\pi_0(\Theta^0) - \epsilon\right)^3}.
\]

Using similar calculations we have

\[
\frac{f_t(\Theta|D^{(t)}(\omega))}{\pi_t(\Theta|D^{(t)}(\omega))} \geq (1 - \eta) \left[ \frac{\prod_{l=1}^t p_{\Theta_1,\Theta_2,t}(D_l)p_{\Theta_1,t,\Theta_2}(D_l)}{\int \prod_{l=1}^t p_\Theta(D_l)} \right] \left[ \frac{\int \prod_{l=1}^t p_\Theta(D_l)}{\prod_{l=1}^t p_\Theta(D_l)} \right] \frac{\left(\pi_0(\Theta^0) - \epsilon\right)^3}{\left(\pi_0(\Theta^0) + \epsilon\right)^3}.
\]

(21) now gives us

\[
\left[ \frac{\prod_{l=1}^t (p_{\Theta}(D_l) - \epsilon)^3}{\prod_{l=1}^t (p_{\Theta}(D_l) + \epsilon)^3} \right] \\
\leq \left[ \frac{\prod_{l=1}^t p_{\Theta_1,\Theta_2,t}(D_l)p_{\Theta_1,t,\Theta_2}(D_l)}{\int \prod_{l=1}^t p_{\Theta_1,\Theta_2,t}(D_l)p_{\Theta_1,t,\Theta_2}(D_l)} \right] \left[ \frac{\int \prod_{l=1}^t p_\Theta(D_l)}{\prod_{l=1}^t p_\Theta(D_l)} \right] \\
\leq \left[ \frac{\prod_{l=1}^t (p_{\Theta}(D_l) + \epsilon)^3}{\prod_{l=1}^t (p_{\Theta}(D_l) - \epsilon)^3} \right]
\]

Using the condition that \(\sqrt{t}p_{\Theta}(D^{(t)}(\omega))\) as \(t \to \infty\) is bounded away from 0 and \(\infty\) and choosing
$\epsilon, \eta$ small we have

$$\left| \frac{f_t(\Theta|D^{(t)}(\omega))}{\pi_t(\Theta|D^{(t)}(\omega))} - 1 \right| < \kappa,$$

for all $t > t_2$, $\omega \in \Omega$. Finally,

$$\int |\pi_t(\Theta) - f_t(\Theta)| \leq \int_{N_{\epsilon,\eta}} |\pi_t(\Theta) - f_t(\Theta)| + \int_{N_{\epsilon,\eta}} |\pi_t(\Theta) - f_t(\Theta)|$$

$$\leq \int_{N_{\epsilon,\eta}} |\pi_t(\Theta) - f_t(\Theta)| + 2\eta$$

$$\leq \pi_t(N_{\epsilon,\eta}) \kappa + 2\eta < \kappa + 2\eta.$$
B Accuracy Plots

<table>
<thead>
<tr>
<th></th>
<th>Accuracy over time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linear regression</strong></td>
<td>![Linear regression Accuracy Plot]</td>
</tr>
<tr>
<td><strong>One-way Anova</strong></td>
<td>![One-way Anova Accuracy Plot]</td>
</tr>
</tbody>
</table>

Figure 5: Parameter accuracy plotted over time as defined in (2) for the motivating examples of Section 3.1. Row #1: accuracy for representative regression coefficients $\beta_j$ and $\sigma^2$ in the linear regression example. Row #2: accuracy for representative group means $\zeta_j$, along with hierarchical parameters $\mu$, $\tau^2$ and $\sigma^2$ for the one-way Anova model.

References


Statistica Sinica, 23, 119-143.


