MTH422: PROJECT REPORT

Stochastic Gradient MCMC for Non Linear State Space Models

Supervisor:

Prof. Arnab Hazara

Group members:

Viral Chitlangia (221198)
Aniruddh Pramod (210142)
Raghav Govind (220852)
Sai Praneeth Donthu (210900)

Abstract

State space models (SSMs) provide a flexible framework for modeling time series with latent dynamics, but Bayesian inference in nonlinear or non-Gaussian SSMs can be quite computationally challenging. Particle filtering methods suffer from high variance and scalability issues, especially for long sequences. Stochastic Gradient MCMC (SG-MCMC) offers a scalable alternative by replacing full-data gradients with stochastic estimates, though naive sub-sampling introduces bias due to temporal dependencies. This is the solution proposed by Aicher et. al in the paper titled, 'Stochastic Gradient MCMC for Nonlinear State Space Models'. In this project, we reproduce the buffered SG-MCMC algorithm, which mitigates bias using a local buffer around each data subsequence. We implement particle-based gradient estimators and demonstrate their performance across three model classes: the Linear Gaussian SSM (LGSSM), Stochastic Volatility Model (SVM), and Generalized Autoregressive Conditional Heteroscedasticity (GARCH). Our experiments validate the geometric error decay with buffer size and highlight the trade-offs between computation and gradient accuracy. These results support buffered SG-MCMC as a practical approach for scalable Bayesian inference in nonlinear state space models.

1 Introduction

Non-linear *state space models* (SSMs) augment an observed time series with a latent state sequence, inducing a Markov Chain dependence structure. This allows for the effective modelling of rich dynamical patterns in engineering, epidemiology, and finance. While particle filters enable Bayesian parameter estimation in these models, their cost grows linearly with the length of the sequence T and also requires a linearly increasing number of particles to control bias.

Stochastic gradient Markov chain Monte Carlo (SG-MCMC) methods replace full-data gradients with stochastic estimates on subsets of data. Naive subsampling typically breaks down temporal dependencies and introduces bias in SSMs. To correct for this, buffered stochastic gradients re-introduce a short context or buffer around each subsequence, which is claimed to control the bias. In this project, we reproduce the particle-based buffered SG-MCMC framework proposed by the authors, derive its theoretical error bounds, and demonstrate its performance on a synthetic time series.

2 Background

In this section we review (i) Nonlinear State Space Models, (ii) Particle Filtering and Smoothing, and (iii) Stochastic Gradient MCMC with Buffered Gradients for SSMs.

2.1 Nonlinear State Space Models for Time Series

State space models are a class of discrete-time bivariate stochastic processes consisting of a latent state process $X_{1:T}$ driving the observed process $Y_{1:T}$. Formally, with parameters θ and prior $X_0 \sim \nu(x_0 \mid \theta)$, where $\theta \in \Theta$ are the set of parameters, the generative model for X and Y is:

$$X_t \mid (X_{t-1} = x_{t-1}, \theta) \sim p(x_t \mid x_{t-1}, \theta), \tag{1}$$

$$Y_t \mid (X_t = x_t, \theta) \sim p(y_t \mid x_t, \theta). \tag{2}$$

Here $p(x_t \mid x_{t-1}, \theta)$ is the *transition density* and $p(y_t \mid x_t, \theta)$ is the *emission density*.

Inference of the model parameters θ, can be done using the *score function*, the gradient of the marginal log-likelihood, $\nabla_\theta \log p(y_{1:T} \mid \theta)$. Using the score function, the log-likelihood can be maximized iteratively

using gradient ascent algorithms or any other optimization methods. We calculate the score function by using Fischers' identity

$$\nabla_\theta \log p(y_{1:T} \mid \theta) = \mathbb{E}_{X|Y,\theta}\big[\nabla_\theta \log p(X_{1:T}, Y_{1:T} \mid \theta)\big] = \sum_{t=1}^{T} \mathbb{E}_{X|Y,\theta}\big[\nabla_\theta \log p(X_t, Y_t \mid X_{t-1}, \theta)\big]. \tag{3}$$

When $p(X_{1:T} \mid Y_{1:T}, \theta)$ is not available analytically, we approximate these expectations via particle methods.

2.1.1 Particle Filtering and Smoothing

Particle filtering methods enable an empirical estimation of the expected value of a function $H(X_{1:T})$ under the posterior distribution $p(x_{1:T} \mid y_{1:T}, \theta)$. This approach involves drawing N samples, referred to as particles $\{x_t^{(i)}\}_{i=1}^N$, and computing corresponding importance weights $\{w_t^{(i)}\}_{i=1}^N$ in a recursive manner over time. To ensure representative sampling, these particles and their weights are iteratively refined through a process known as sequential importance resampling.

Particle filters maintain N weighted particles $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^N$. At each t:

1. *Resample:* draw ancestor indices
$$a_i \sim \text{Categorical}(w_{t-1}^{(i)}).$$

2. *Propagate:* sample from a proposal density $q(\cdot|\cdot)$
$$x_t^{(i)} \sim q\big(x_t \mid x_{t-1}^{(a_i)}, y_t, \theta\big).$$

3. *Weight:* update and normalize
$$w_t^{(i)} \;\propto\; \frac{p(y_t \mid x_t^{(i)}, \theta)\, p(x_t^{(i)} \mid x_{t-1}^{(a_i)}, \theta)}{q(x_t^{(i)} \mid x_{t-1}^{(a_i)}, y_t, \theta)}, \quad \sum_{i=1}^{N} w_t^{(i)} = 1. \tag{4}$$

4. *Statistic update:* for additive statistics $H_T = \sum_{t=1}^{T} h_t(x_t, x_{t-1})$, maintain
$$H_t^{(i)} = H_{t-1}^{(a_i)} + h_t\big(x_t^{(i)}, x_{t-1}^{(a_i)}\big).$$

The auxiliary variables $\{a_i\}_{i=1}^N$ denote the indices of the ancestor particles $\{x_t^{(i)}\}_{i=1}^N$ selected at time t. These indices help trace the evolutionary paths of particles across time steps. The resampling procedure defined in step (i) outlines how new particles are generated by selecting from existing ones based on their importance weights.

Resampling is performed at each iteration to address the issue of weight degeneracy. This issue arises when a small number of particles dominate the total weight, rendering most particles ineffective. To combat this, several resampling strategies have been proposed, including stratified and residual resampling, among others.

A specific case occurs when the proposal distribution $q(x_t \mid x_{t-1}, y_t, \theta)$ is chosen to be the same as the system's transition model $p(x_t \mid x_{t-1}, \theta)$. In this scenario, the particle filter simplifies, and the recursive update of importance weights reduces to $w_t^{(i)} \propto \hat{p}(y_t \mid x_t^{(i)}, \theta)$.

In situations where the target function $H(X_{1:T})$ can be expressed as a sum of pairwise functions, $H(X_{1:T}) = \sum_{t=1}^{T} h_t(x_t, x_{t-1})$, such as in gradient-based identities, the filtering algorithm can be adapted to maintain a running sum $H_t = \sum_{s=1}^{t} h_s(x_s, x_{s-1})$. This allows efficient tracking of relevant quantities throughout the filtering process.

Algorithm 1 Particle Filter

Input: number of particles, N, pairwise statistics, $h_{1:T}$, observations $y_{1:T}$, proposal density q
Draw $x_0^{(i)} \sim \nu(x_0|\theta)$, set $w_0^{(i)} = \frac{1}{N}$, and $H_0^{(i)} = 0\ \forall i$
for t = 1, . . . , T **do**
    Resample ancestor indices $\{a_1, \ldots, a_N\}$
    Propagate particles $x_t^{(i)} \sim q(\cdot|x_{t-1}^{(a_i)}, y_t, \theta)$
    Update each $w_t^{(i)}$ according to (3)
    Update statistics $H_t^{(i)} = H_{t-1}^{(a_i)} + h_t(x_t^{(i)}, x_{t-1}^{(a_i)})$
end for
Return $H = \sum_{i=1}^{N} w_T^{(i)} H_T^{(i)}$

2.2 Stochastic Gradient MCMC

A widely used approach for performing scalable Bayesian inference on large datasets is the Stochastic Gradient Markov Chain Monte Carlo (SGMCMC) framework. Given a prior distribution $p(\theta)$, the goal is to generate samples θ from the posterior $p(\theta \mid y) \propto p(y \mid \theta)p(\theta)$. Gradient-based MCMC methods achieve this by simulating a stochastic differential equation (SDE) that incorporates the gradient of the log-likelihood, denoted as $g_\theta = \nabla_\theta \log p(y \mid \theta)$. The posterior distribution is designed to be the stationary distribution of this SDE.

To make the computation feasible for large datasets, SGMCMC methods approximate the full-data gradient with a stochastic estimate $\hat{g}_\theta$, obtained using mini-batches of data. This significantly reduces computational cost.

Among various SGMCMC techniques, the Stochastic Gradient Langevin Dynamics (SGLD) algorithm is one of the most prominent methods.

$$\theta^{(k+1)} \leftarrow \theta^{(k)} + \epsilon^{(k)} \left(\hat{g}_\theta + \nabla_\theta \log p(\theta)\right) + N(0, 2\epsilon^{(k)}),$$

Here, $\epsilon^{(k)}$ denotes the step size, and θ_1 is the initial point of the chain. If $\hat{g}_\theta$ is an unbiased gradient estimator and $\epsilon^{(k)}$ decays appropriately, the distribution of θ converges to the true posterior.

2.2.1 Stochastic Gradients for SSMs

Stochastic Gradients for SSMs

An additional challenge when applying SGMCMC to SSMs is handling the temporal dependence between observations. Based on a subset $\mathcal{S}$ of size S, an unbiased stochastic gradient estimate of (3) is:

$$\sum_{t \in S} \Pr(t \in S)^{-1} \cdot \mathbb{E}_{x|y_{1:T},\theta} \left[\nabla_\theta \log p(X_t, y_t \mid X_{t-1}, \theta)\right] \tag{5}$$

Although (5) is a sum over S terms, it requires taking expectations with respect to $p(x|y_{1:T}, \theta)$, which requires processing the full sequence $y_{1:T}$. One approach to reduce computation is to randomly sample $\mathcal{S}$ as a contiguous subsequence $\mathcal{S} = \{s+1, \ldots, s+S\}$ and approximate (5) using only y_s:

$$\sum_{t \in \mathcal{S}} \Pr(t \in \mathcal{S})^{-1} \cdot \mathbb{E}_{x|y_{\mathcal{S}},\theta} \left[\nabla_\theta \log p(X_t, y_t|X_{t-1}, \theta)\right]. \tag{6}$$

However, 6 is biased because the expectation over the latent states x_s is conditioned only on y_s rather than $y_{1:T}$.

To control the bias in stochastic gradients while also avoiding accessing the full sequence, previous work on SGMCMC for SSMs proposed buffered stochastic gradient:

$$\hat{g}_\theta(S, B) = \sum_{t \in \mathcal{S}} \frac{\mathbb{E}_{x|y_{\mathcal{S}^*},\theta} \left[\nabla_\theta \log p(X_t, y_t | X_{t-1}, \theta)\right]}{\Pr(t \in \mathcal{S})}, \tag{7}$$

where $\mathcal{S}^* = \{s+1-B, \ldots, s+S+B\}$ is the buffered subsequence such that $\mathcal{S} \subset \mathcal{S}^* \subset \{1, \ldots, T\}$. When the buffer extends outside of the original subsequence then we can extend the model to $\{1-B, \ldots, T+B\}$ and assume the observations y_t outside of $\{1, \ldots, T\}$ are missing. In practice, we will truncate S^* by intersecting it with $\{1, \ldots, T\}$.

The unbiased gradient estimate, which conditions on all data (5), is $\hat{g}_\theta(S, T)$ and the gradient for no buffering (7) is $\hat{g}_\theta(S, 0)$. As B increases from 0 to T, the estimator $\hat{g}_\theta(S, B)$ provides reduced bias.

3 Method

We now present a particle-based buffered stochastic gradient estimator for nonlinear state-space models, and show how to integrate it into stochastic gradient Langevin dynamics.

3.1 Buffered Stochastic Gradient Estimates for Nonlinear SSMs

Select a contiguous subsequence

$$\mathcal{S} = \{s+1, \ldots, s+S\}$$

and extend it with a buffer of size B:

$$\mathcal{S}^* \;=\; \{\, s+1-B, \ldots, s+S+B\} \;\cap\; \{1, \ldots, T\}.$$

Our target is the buffered gradient

$$\widehat{g}_\theta(S, B) = \sum_{t \in \mathcal{S}} \Pr(t \in \mathcal{S})^{-1}\, \mathbb{E}_{X_{1:T}|y_{\mathcal{S}^*},\theta}\Big[\nabla_\theta \log p(X_t, Y_t \mid X_{t-1}, \theta)\Big].$$

We approximate this expectation over $p(x|y_{\mathcal{S}^*}, \theta)$ by running the bootstrap particle filter with $N = 1000$ particles. Let this approximation $g_\theta^{PF}(S, B, N)$. The H term required in the particle filter method is given below in this particular case.

$$H \;=\; \sum_{t \in \mathcal{S}^*} h_t(x_t, x_{t-1}), \tag{8}$$

where

$$h_t(x_t, x_{t-1}) = \begin{cases} \dfrac{\nabla_\theta \log p(x_t, y_t \mid x_{t-1}, \theta)}{\Pr(t \in \mathcal{S})}, & t \in \mathcal{S}, \\ 0, & t \notin \mathcal{S}. \end{cases} \tag{9}$$

Although $h_t = 0$ outside S, including the buffer indices in the particle filter reduces bias.

3.2 Buffered PF–SGLD

We plug $g_\theta^{\mathrm{PF}}(S, B, N)$ into stochastic gradient Langevin dynamics. The full algorithm is provided below in 2 Each iteration requires $O\big((S+2B)\,N\big)$ work, balancing bias reduction (via B) against computational cost.

Algorithm 2 Buffered PF-SGLD

Input: Data $y_{1:T}$, Initial $\theta^{(0)}$, Stepsize ϵ, Subsequence Size S, Buffer Size B, Particle Number N
for $k = 1, 2, \ldots, K$ **do**
    Sample $\mathcal{S} = \{s+1, \ldots, s+S\}$
    Set $\mathcal{S}^* = \{s+1-B, \ldots, s+B\}$
    Calculate g_θ^{PF} over $\mathcal{S}^*$ using Algorithm 1 on (9)
    Set $\theta^{(k+1)} \leftarrow \theta^{(k)} + \epsilon \cdot (g_\theta^{PF} + \nabla \log p(\theta)) + N(0, 2\epsilon)$
end for
Return $\theta^{(K+1)}$

4 Experiments

The first task is to verify the empirical test for the bias of the particle buffered gradient estimator g_θ^{PF} on synthetic data for fixed θ. We then evaluate the performance of the proposed Algorithm 2 on synthetic data.

4.1 Models

Three models are considered for the experiments: (i) the linear Gaussian SSM (LGSSM); (ii) Stochastic Volatility Model (SVM); and (iii) Generalized Auto Regressive Conditional Heteroskedasticity Model (GARCH). These have been chosen to assess the impact of the particle filter, test on non-Gaussian emissions and to evaluate the model on non-linear latent transitions respectively.

4.2 Stochastic Gradient Bias

We compare the error of the stochastic gradient estimates using a buffered subsequence with size $S = 16$ while varying B and N on synthetic data from each model. The length of the synthetic data generated in each case is $T = 256$. The bias of the particle buffered stochastic gradient is averaged over 1000 replications. Buffer size is varied from $B \in [0, 16]$, the subsequence size is varied from $[1, T]$ and the number of samples is varied from $N \in \{100, 1000, 10000\}$

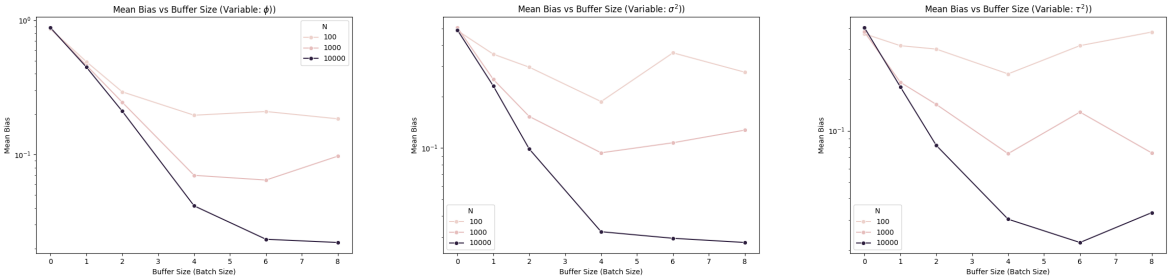
4.2.1 LGSSM

The *linear Gaussian SSM* (LGSSM) is given by -

$$X_t|(X_{t-1} = x_{t-1}, \theta) \sim N(x_t|\phi x_{t-1}, \sigma^2) \tag{10}$$

$$Y_t|(X_t = x_t, \theta) \sim N(y_t|x_t, \tau^2) \tag{11}$$

Here, $\nu_0(x_0) = N(x_0|0, \frac{\phi^2}{1-\sigma^2})$ and parameters $\theta = (\phi, \sigma, \tau)$. The transition and emission distributions here are both Gaussian and log-concave in x. The goal is to show geometrically decreasing buffering error with increasing buffer size B. This behaviour can be clearly seen in the following log-log plots in Figure 1.



(a) ϕ (b) σ^2 (c) τ^2

Figure 1: Mean Bias against the Buffer Size for the LGSSM Model

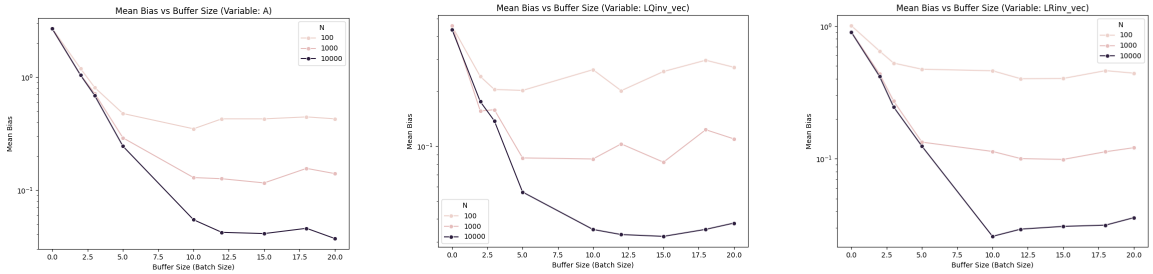
4.2.2 Stochastic Volatility Model

The *Stochastic Volatility Model* (SVM) is given by -

$$X_t|(X_{t-1} = x_{t-1}, \theta) \sim N(x_t|\phi x_{t-1}, \sigma^2) \tag{12}$$

$$Y_t|(X_t = x_t, \theta) \sim N(y_t|0, \exp(x_t)\tau^2) \tag{13}$$

Here, $\nu_0(x_0) = N(x_0|0, \frac{\phi^2}{1-\sigma^2})$ and parameters $\theta = (\phi, \sigma, \tau)$. The transition and emission distributions here are again log-concave in x. Thus, we can observe similar geometrically decay behavior in the buffering error with increasing buffer size B in the plots in Figure 2.



(a) ϕ (b) LQ Inverse (c) LR Inverse

Figure 2: Buffer Size vs Mean Bias (Log Scaled) for SVM Model

4.2.3 GARCH Model

Finally we look at the noisy GARCH(1, 1) model -

$$X_t|(X_{t-1} = x_{t-1}, \sigma_t^2, \theta) \sim N(x_t|0, \sigma_t^2) \tag{14}$$

$$\sigma_t^2(x_{t-1}, \sigma_{t-1}^2, \theta) = \alpha + \beta x_{t-1}^2 + \gamma \sigma_{t-1}^2 \tag{15}$$

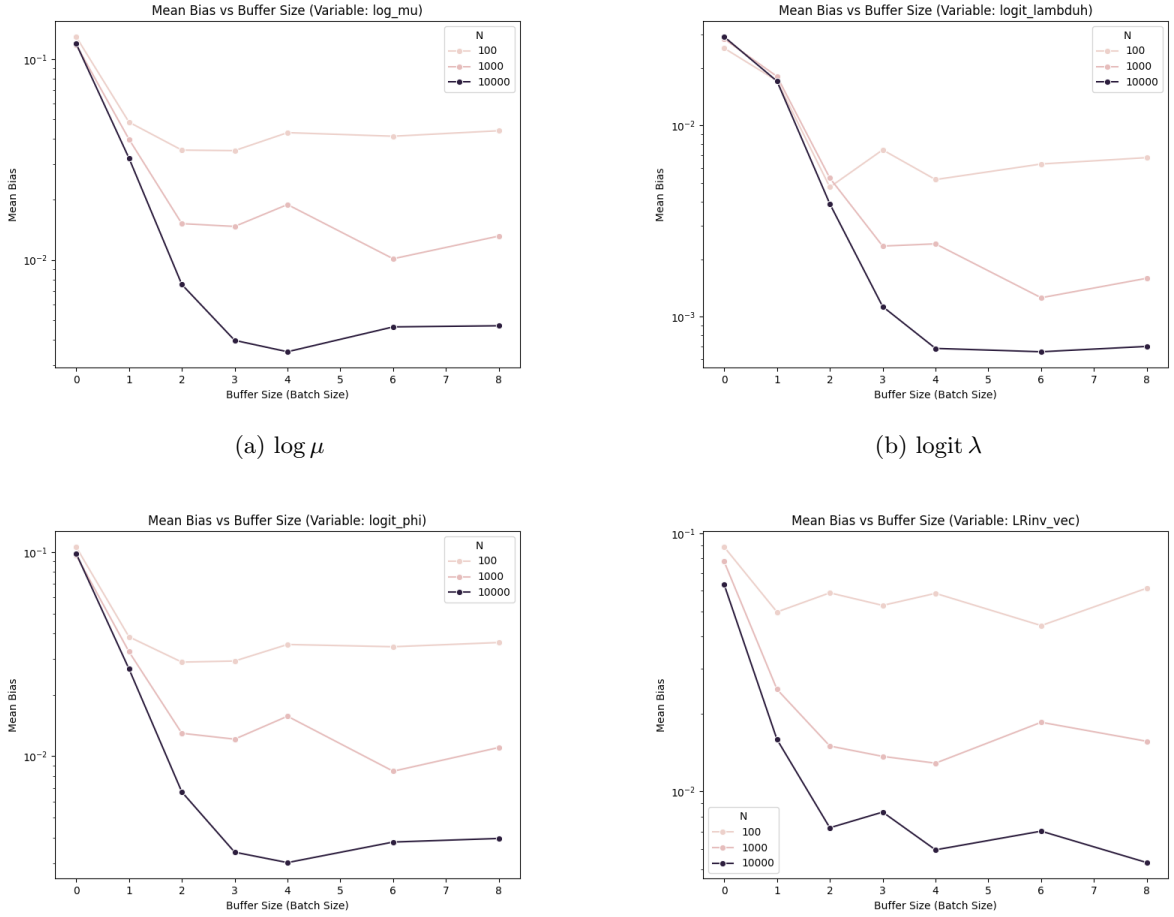
$$Y_t|(X_t = x_t, \theta) \sim N(y_t|x_t, \tau^2) \tag{16}$$

Here, $\nu_0(x_0) = N(x_0|0, \frac{\alpha}{1-\beta-\gamma})$ and parameters $\theta = (\alpha, \beta, \gamma, \tau)$. This model's transition distribution is not log-concave in $(x_t,\ x_{t-1})$, but we can empirically see that buffering helps reduce the gradient error for

the GARCH model. We convert the parameters (α, β, γ) to $(\log \mu, \operatorname{logit} \phi, \operatorname{logit} \lambda)$ as

$$\log \mu = \log \left(\frac{\alpha}{1 - \beta - \gamma} \right)$$
$$\operatorname{logit} \phi = \operatorname{logit} (\beta + \gamma)$$
$$\operatorname{logit} \lambda = \operatorname{logit} \left(\frac{\beta}{\beta + \gamma} \right)$$

The behaviour of the gradient error with increasing buffer size can be seen in the plots in Figure 3.



(a) $\log \mu$

(b) $\operatorname{logit} \lambda$

(c) $\operatorname{logit} \phi$

(d) LR_Inv

Figure 3: Buffer Size vs Mean Bias (Log Scaled) for GARCH Model

4.3 SGLD Experiments

This section of the report conducts experiments on the PF - SGLD method described in the paper, namely the buffered stochastic gradient estimators used in the SGLD sampler.

4.3.1 Kernel Stein Discrepency For SGLD Evaluation

To evaluate the MCMC chains $\{\theta^{(k)}\}_{k=1}^{K}$, instead of using the ESS diagnostic, the Kernel Stein Discrepency (KSD) metric is used since it also penalizes the bias in the chains. Given a sample chain after burn-in and thinning, $\{\theta^{(k)}\}_{k=1}^{\tilde{K}}$, let $\hat{p}(\theta \mid y)$ be the distribution of the parameter samples. Then the Kernel Stein Discrepancy (KSD) between $\hat{p}(\theta \mid y)$ and the posterior distribution $p(\theta \mid y)$ is

$$\text{KSD}(\hat{p}, p) = \sum_{d=1}^{\dim(\theta)} \sqrt{\sum_{k,k'=1}^{\tilde{K}} \frac{K_d^0(\theta^{(k)}, \theta^{(k')})}{\tilde{K}^2}},$$

where

$$K_d^0(\theta, \theta') = \frac{1}{p(\theta \mid y)p(\theta' \mid y)} \nabla_{\theta_d} \nabla_{\theta'_d} \left(p(\theta \mid y) K(\theta, \theta') p(\theta' \mid y) \right),$$

and $K(\cdot, \cdot)$ is the below kernel function:

$$K(\theta, \theta') = \left(1 + \|\theta - \theta'\|_2^2\right)^{-0.5}$$

4.3.2 SGLD on Synthetic Data

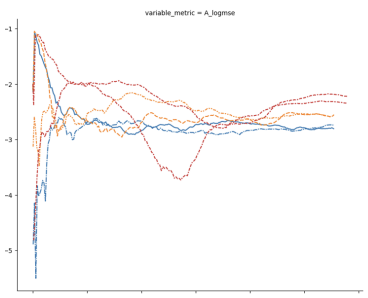
In this and the upcoming sections we conduct experiments with the Particle Filter Buffered Stochastic Gradients in the SGLD sampler. We focus on three sets of synthetic data generated from the LGSSM, SVM and GARCH model respectively. $\hat{g}_\theta(S, B)$ is calculated and used. The exact same model parametrization as in the previous section is used for all three models.

For each model we consider three different values for $\{S, B\}$. **Buffered** is for the pair $(S = 40, B = 10)$, **Full** is the scenario when $S = T$ or the complete data set and **No Buffer** is for $(S = 40, B = 0)$. For each model we consider he number of particles $N = 1000$. The step size value has been selected corresponding to the value used in the paper; which is chosen in accordance to the least KSD value. In each case for each model, different Particle Filters are used which will vastly reduce compute time and will also give results on the dependency of particle filters.

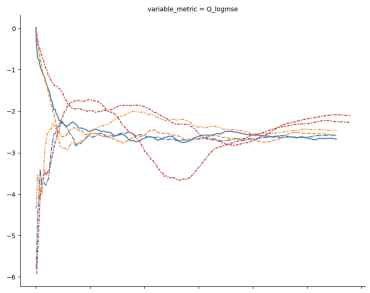
4.3.3 On Synthetic LGSSM data

Running the SGLD sampler with the PF buffered gradients wherever required on the synthetic dataset generated by the LGSSM model, below are the plots for the estimated conjugate posteriors of the different parameters for the LGSSM model when using different particle filters in the Full, Buffer, and No Buffer setting. The plots denote the MSE of the estimated posterior against the true values. Note that the size of the LGSSM dataset is 1000 datapoints. The blue lines are the plots for the POYIADJIS N = 1000 particle filter. The green lines are the plots for the MC N = 100 particle filter. The red lines are the plots for the NEMETH N = 1000 particle filter. The orange lines are the plots for the Kalman filter. The different line types of the plots are for different settings and setups for the particle filters apart from the number of particles.

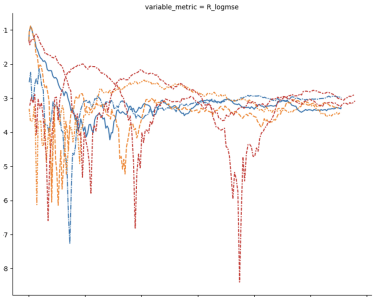
The below plots are for the **BUFFERED** setting:-



(a) LOG MSE vs TIME for A parameter



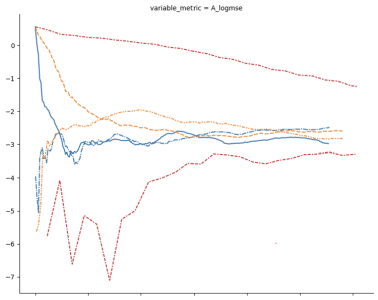
(b) LOG MSE vs TIME for Q parameter



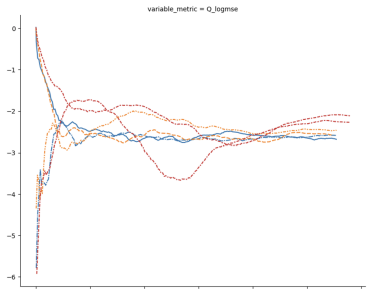
(c) LOG MSE vs TIME for R parameter

Figure 4: Plots of MSE against parameter for LGSSM data with **BUFFERED** setting

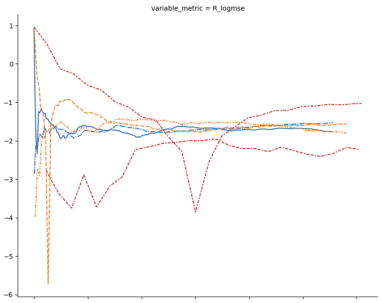
The below plots are for the **FULL** setting:-



(a) LOG MSE vs TIME for A parameter



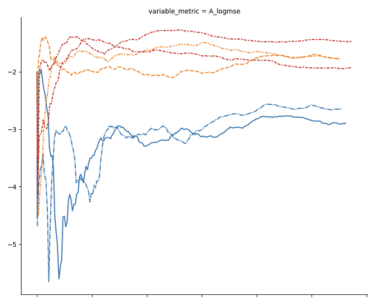
(b) LOG MSE vs TIME for Q parameter



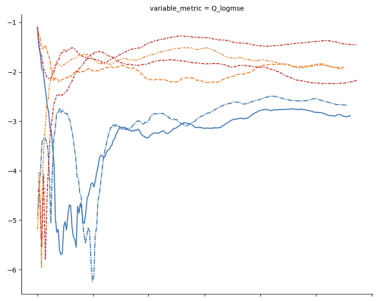
(c) LOG MSE vs TIME for R parameter

Figure 5: Plots of MSE against parameter for LGSSM data with **FULL** setting

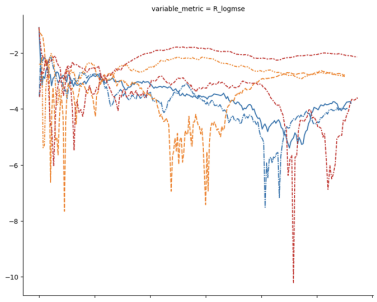
The below plots are for the **NO BUFFERED** setting:-



(a) LOG MSE vs TIME for A parameter



(b) LOG MSE vs TIME for Q parameter



(c) LOG MSE vs TIME for R parameter

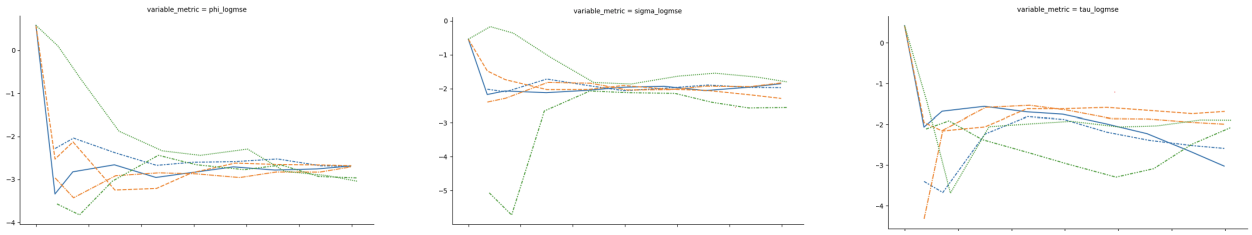
Figure 6: Plots of MSE against parameter for LGSSM data with **NO BUFFERED** setting

4.3.4 On Synthetic SVM data

Running the SGLD sampler with the PF buffered gradients wherever required on the synthetic dataset generated by the SVM model, below are the plots for the estimated conjugate posteriors of the different parameters for the SVM model when using different particle filters in the Full, Buffer, and No Buffer setting.

The plots denote the MSE of the estimated posterior against the true values. Note that the size of the GARCH dataset is 1000 datapoints. The blue lines are the plots for the POYIADJIS $N = 1000$ particle filter. The green lines are the plots for the PARIS $N = 100$ particle filter. The orange lines are the plots for the NEMETH $N = 1000$ particle filter. The different line types of the plots are for different settings and setups for the particle filters apart from the number of particles.

The below plots are for the **BUFFERED** setting:-



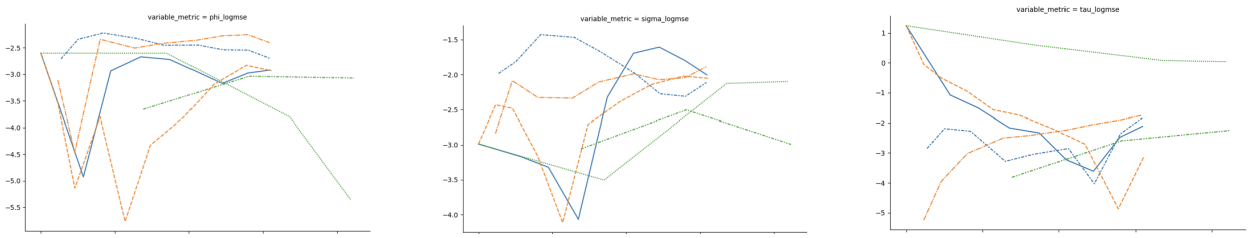
(a) LOG MSE vs TIME for phi parameter

(b) LOG MSE vs TIME for sigma parameter

(c) LOG MSE vs TIME for tau parameter

Figure 7: Plots of MSE against parameter for SVM data with **BUFFERED** setting

The below plots are for the **FULL** setting:-



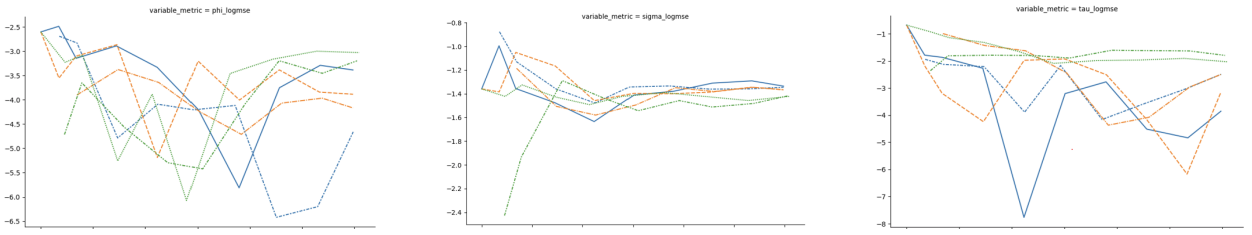
(a) LOG MSE vs TIME for phi parameter

(b) LOG MSE vs TIME for sigma parameter

(c) LOG MSE vs TIME for tau parameter

Figure 8: Plots of MSE against parameter for SVM data with **FULL** setting

The below plots are for the **NO BUFFERED** setting:-



(a) LOG MSE vs TIME for phi parameter

(b) LOG MSE vs TIME for sigma parameter

(c) LOG MSE vs TIME for tau parameter

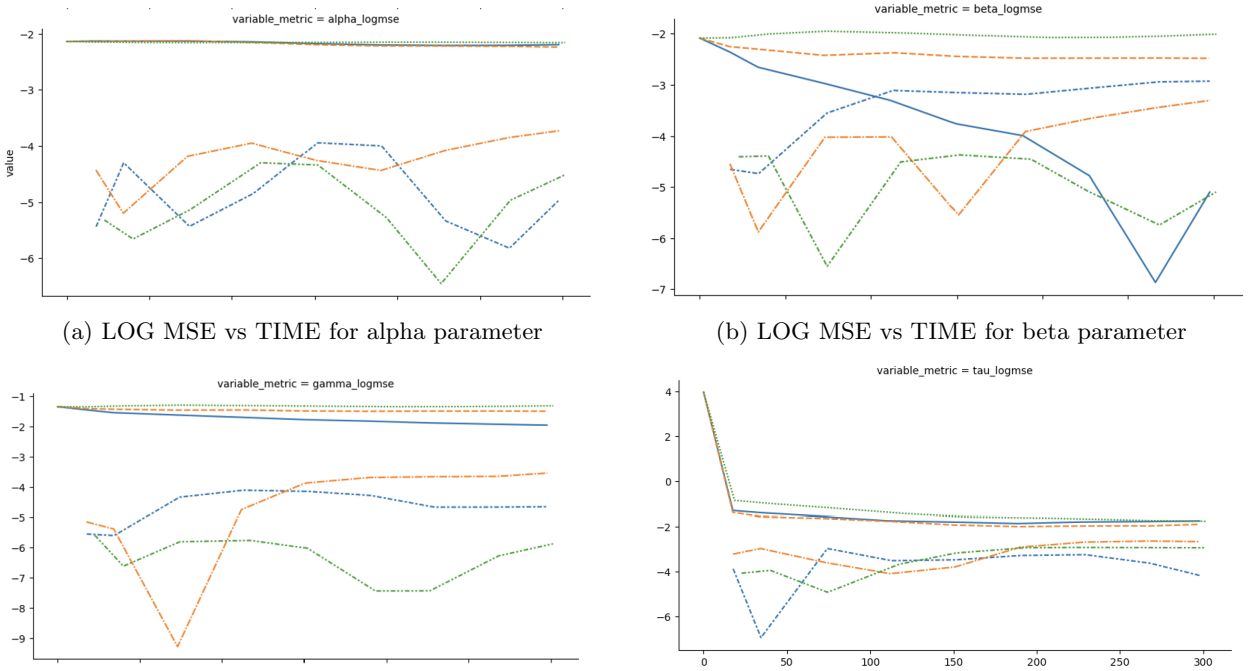
Figure 9: Plots of MSE against parameter for SVM data with **NO BUFFERED** setting

4.3.5 On Synthetic GARCH data

Running the SGLD sampler with the PF buffered gradients wherever required on the synthetic dataset generated by the GARCH model, below are the plots for the estimated conjugate posteriors of the different

parameters for the GARCH model when using different particle filters in the Full, Buffer, and No Buffer setting. The plots denote the MSE of the estimated posterior against the true values. Note that the size of the GARCH dataset is 1000 datapoints. The blue lines are the plots for the POYIADJIS $N = 1000$ particle filter. The green lines are the plots for the PARIS $N = 100$ particle filter. The orange lines are the plots for the NEMETH $N = 1000$ particle filter. The different line types of the plots are for different settings and setups for the particle filters apart from the number of particles. The KSD values for each of the parameters were calculated for each of the parameters but since the grid search method for identifying the best setting was computationally expensive to run, we ended up using the settings used in the paper.

The below plots are for the **BUFFERED** setting:-



(a) LOG MSE vs TIME for alpha parameter

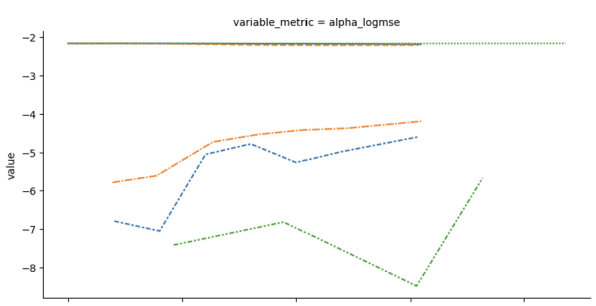
(b) LOG MSE vs TIME for beta parameter

(c) LOG MSE vs TIME for gamma parameter

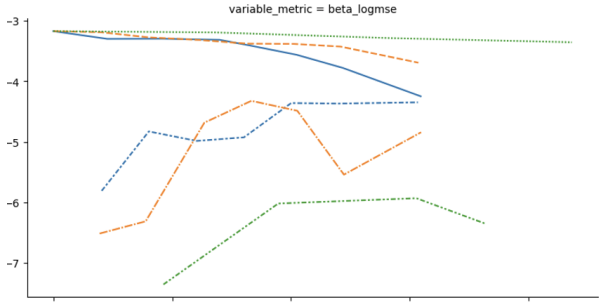
(d) LOG MSE vs TIME for tau parameter

Figure 10: Plots of MSE against parameter for GARCH data with **BUFFERED** setting

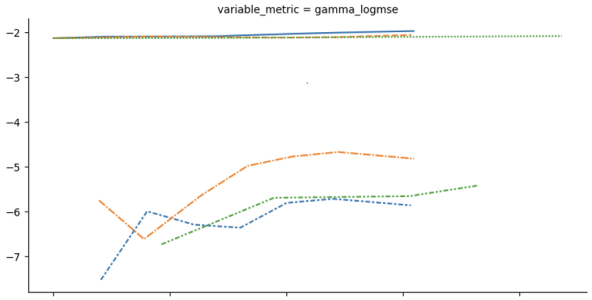
The below plots are for the **FULL** setting:-



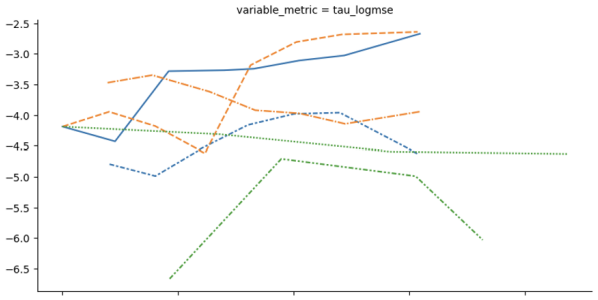
(a) LOG MSE vs TIME for alpha parameter



(b) LOG MSE vs TIME for beta parameter



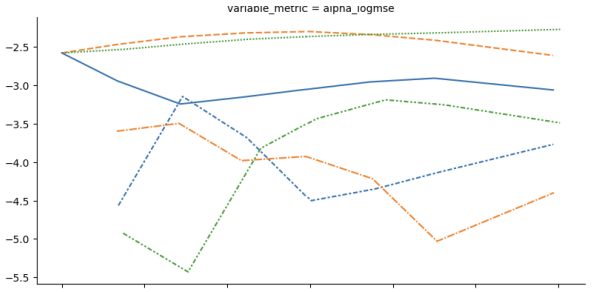
(c) LOG MSE vs TIME for gamma parameter



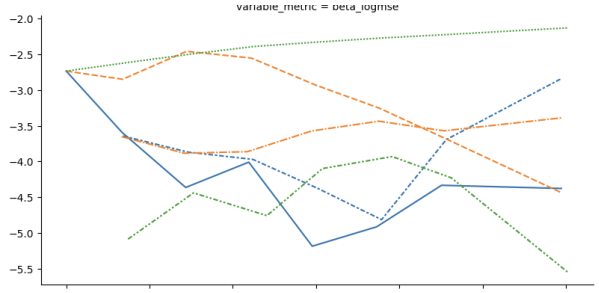
(d) LOG MSE vs TIME for tau parameter

Figure 11: Plots of MSE against parameter for GARCH data with **FULL** setting

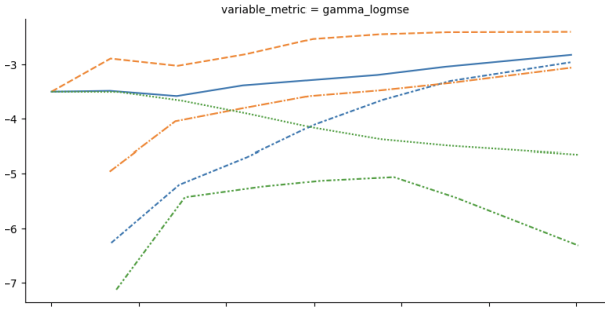
The below plots are for the **NO BUFFER** setting:-



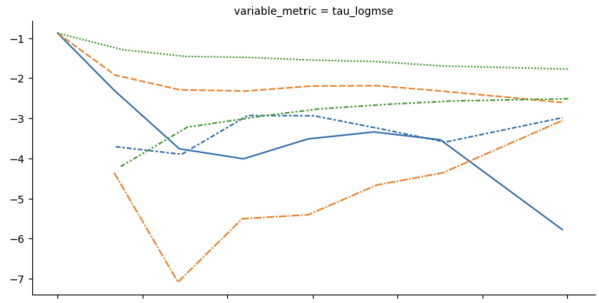
(a) LOG MSE vs TIME for alpha parameter



(b) LOG MSE vs TIME for beta parameter



(c) LOG MSE vs TIME for gamma parameter



(d) LOG MSE vs TIME for tau parameter

Figure 12: Plots of MSE against parameter for GARCH data with **NOBUFF** setting

4.4 SGLD on Real Life Exchange Data

The paper then proceeds to fit the SVM and GARCH models to a EUR-USD dataset at the minute resolution over the timescale of a year. This data consists of $350,000$ values. Unfortunately we found that the models take extremely long in the fitting process even with relaxed parameter settings of $N = 1000$ particles, whereas the actual results were produced using $N = 10000$ particles. With relaxed parameters, we were not able to complete the process even after multiple days of execution time, and considering that the actual results would likely take 10 times the execution time, we opted not to validate the results of this section.

5 Contributions

- Viral Chitlangia - Updated code to work on Synthetic Data generated by LGSSM, SVM and Garch Models. Created the plots to compare for various models.
- Aniruddh Pramod - Sourced the paper, updated code to work with a more modern version of NumPy, set up, prepared and ran the driver scripts and experiment scripts for the Synthetic Data Experiments for all the models. Worked on the report.
- Raghav Govind - Understood the mathematical principles used in the plots and the methodology.
- Sai Praneeth Donthu - Ran the scripts for plotting all the figures for the Synthetic Data and tabulation of Simulation results and KSD values. Also, understood the math behind the method.