Generalized Posterior Calibration via Sequential Monte Carlo Sampler

Masahiro Tanaka
gspddlnit45@toki.waseda.jp
Faculty of Economics, Fukuoka University,
Fukuoka, Fukuoka, Japan

ABSTRACT

For robustness toward model misspecification, the generalized posterior inference approach modifies the likelihood term by raising it to the power of a learning rate, thereby adjusting the spread of the posterior. This paper proposes a computationally efficient strategy for selecting an appropriate learning rate. The proposed approach builds upon the generalized posterior calibration (GPC) algorithm introduced by Syring and Martin (2019) [Biometrika, Volume 106, Issue 2, pp. 479-486], which is designed to select the learning rate to achieve the nominal frequentist coverage. This algorithm, which evaluates the coverage probability based on bootstrap samples, suffers from high computational costs due to the need for repeated posterior simulations for bootstrap samples. To address this limitation, the study proposes an algorithm that combines elements of the GPC algorithm with the sequential Monte Carlo (SMC) sampler. By leveraging the similarity between the learning rate in generalized posterior inference and the inverse temperature in SMC sampling, the proposed algorithm efficiently calibrates the posterior distribution with less computational cost. For demonstration, the proposed algorithm was applied to several statistical learning models.

CCS CONCEPTS

• **Mathematics of computing** → **Probabilistic algorithms**; **Probabilistic inference problems**; **Bayesian computation**; • **Computing methodologies** → **Uncertainty quantification**.

KEYWORDS

generalized posterior, Gibbs posterior, generalized posterior calibration, sequential Monte Carlo sampler

1 INTRODUCTION

As the amount of available data and the complexity of workhorse models increase, the need for robustness in statistical learning becomes more pressing. Given the trend, this paper focuses on two strands of robust Bayesian methods. The first strand is generalized Bayesian inference [21, 24]. Let $\mathcal{D}$ denote a dataset with a sample size N and consider it to be independent, $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^N$. In standard Bayesian inference, the posterior of a K-dimensional unknown parameter vector $\boldsymbol{\theta}$ is composed of a likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ and prior $p(\boldsymbol{\theta})$,

$$\pi(\boldsymbol{\theta}) \propto p(\mathcal{D}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}),$$

In contrast, the generalized Bayesian inference is based on a generalized posterior that is obtained by equipping a learning rate η (> 0) (also called a scaling parameter) with the following likelihood:

$$\pi_\eta^*(\boldsymbol{\theta}) \propto p(\mathcal{D}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})$$

By setting $\eta < 1$, we can increase the spread of the posterior, making the inference robust to model misspecifications (e.g., [7, 22, 23, 31]). The second class of robust Bayesian methods is Gibbs posterior inference [1, 3, 14, 28, 34, 35]. This approach formulates an inferential problem using a generic loss function $r_i(\boldsymbol{\theta}; \mathcal{D}_i)$ instead of a probabilistic model:

$$\pi_\eta^*(\boldsymbol{\theta}) \propto \exp\{-Nr(\boldsymbol{\theta}; \mathcal{D})\}^\eta\, p(\boldsymbol{\theta}),$$

where $r(\boldsymbol{\theta}; \mathcal{D})$ denotes an empirical risk function defined as

$$r(\boldsymbol{\theta}; \mathcal{D}) = \frac{1}{N}\sum_{i=1}^N r_i(\boldsymbol{\theta}; \mathcal{D}_i).$$

The algorithm proposed in the paper can be applied to both Gibbs and generalized posterior inferences. Subsequently, $q(\boldsymbol{\theta}; \mathcal{D})$ denotes a likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ or pseudo-likelihood $\exp\{-Nr(\boldsymbol{\theta}; \mathcal{D})\}$,

$$\pi_\eta^*(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta}; \mathcal{D})^\eta\, p(\boldsymbol{\theta}). \tag{1}$$

There are several approaches to select η ([8, 11, 20, 27]), and each with a different focus[1]. We intend to improve the generalized posterior calibration (GPC) algorithm proposed by Syring and Martin [27]. In the GPC, the coverage probability under a specific value of η is evaluated using bootstrap samples, and η is numerically chosen to achieve the nominal frequentist coverage probability. A notable limitation of the GPC is its computational cost. We must repeatedly run a posterior simulator for bootstrap samples until convergence.

The contribution of this study is that it develops a new computational strategy for choosing the learning rate η. Essentially, the algorithm we propose is a fusion of the GPC and sequential Monte Carlo (SMC) samplers [4, 5]. SMC samplers are a class of algorithms that apply importance sampling to intermediate distributions that bridge from a prior to a posterior. The learning rate in generalized posterior inference and the inverse temperature in SMC samplers play the same role–powering the likelihood. By exploiting this similarity, we transform the target distribution gradually to achieve the target credible/confidence level. Indeed, Syring and Martin [27] mentioned a related idea in their supplementary material.[2] However, to the best knowledge of the author, no practical implementation of this idea has been studied. Furthermore, it is ineffective to apply importance sampling directly to the problem because its quality critically depends on the disparity between the proposal and target distributions (e.g., [19]); when the consecutive learning rates, i.e., the proposal and target distributions, are not

[1]See [33] for a comparison.
[2]"In our implementation of the algorithm, the posterior is sampled every time ω [note by the author: learning rate] is updated. However, it may be faster to sample the posterior M times for ω_0 and subsequently use importance sampling to update the posterior samples each time ω is updated." (p. 2 of the Supplementary Material).

close enough, importance sampling would not work well. Therefore, the use of an SMC sampler is of critical importance.

The remainder of this paper is structured as follows. Section 2 introduces the proposed algorithm. In Section 3, we apply the algorithm to synthetic and real data for demonstration. Section 4 concludes the paper.

2 METHOD

2.1 Generalized posterior calibration

This section describes the GPC [27]. We refer to it as the GPC-MCMC to distinguish it from the proposed algorithm. Let $C_\alpha^\eta(\mathcal{D})$ denote the generalized posterior $100(1-\alpha)\,\%$ credible set for $\boldsymbol{\theta}$ with η. The coverage probability function is represented as follows:

$$c_\alpha(\eta|\mathbb{P}) = \mathbb{P}\left\{\boldsymbol{\theta}^\dagger(\eta) \in C_\alpha^\eta(\mathcal{D})\right\},$$

where $\boldsymbol{\theta}^\dagger(\eta)$ denotes the Kullback-Leibler minimizer in the model obtained with η. As the true data distribution $\mathbb{P}$ is unknown, we replace it with the empirical distribution $\mathbb{P}_N$,

$$c_\alpha(\eta|\mathbb{P}_N) = \mathbb{P}_N\left\{\hat{\boldsymbol{\theta}}(\eta) \in C_\alpha^\eta(\mathcal{D})\right\},$$

where $\hat{\boldsymbol{\theta}}(\eta)$ represents a point estimate of $\boldsymbol{\theta}$ obtained with η. Even with this modification, $c_\alpha(\eta|\mathbb{P}_N)$ cannot be evaluated because enumeration of all N^N possible with-replacement samples from $\mathcal{D}$ is needed. Therefore, we approximate $\mathbb{P}_N$ using a bootstrap method. With B bootstrap samples $\left\{\breve{\mathcal{D}}^{[b]}\right\}_{b=1}^B$, the coverage probability can be estimated as

$$\hat{c}_\alpha(\eta|\mathbb{P}_N) = \frac{1}{B}\sum_{b=1}^{B}\mathbb{I}\left\{\hat{\boldsymbol{\theta}}(\eta) \in C_\alpha^\eta\left(\breve{\mathcal{D}}^{[b]}\right)\right\},$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function. η is chosen by solving $\hat{c}_\alpha(\eta|\mathbb{P}_N) = 1-\alpha$ via a stochastic approximation [26]. At the sth iteration, a single step of the stochastic approximation recursion is given as

$$\eta_{(s+1)} \longleftarrow \eta_{(s)} + \varsigma_l\left[\hat{c}_\alpha\left(\eta_{(s)}|\mathbb{P}_N\right) - (1-\alpha)\right], \tag{2}$$

where $\{\varsigma_l\}$ denotes a non-increasing sequence such that $\sum_l \varsigma_l = \infty$ and $\sum_l \varsigma_l^2 < \infty$. While Syring and Martin [27] specify $\varsigma_s = s^{-0.51}$ in their study, we used a variant of Keston's [16] rule: $\varsigma_s(l) = l^{-0.51}$ where l increases by 1 only when there is a directional change in the trajectory of $\eta_{(s)}$ and $\hat{c}_\alpha\left(\eta_{(s)}|\mathbb{P}_N\right) < 1$. This modification shortens the convergence time significantly. Let R denote the number of posterior draws used for analysis after discarding initial draws as warmup. Figure 1 summarizes the GPC-MCMC.

Figure 1: Generalized posterior calibration

`input:` observed dataset $\mathcal{D}$, target kernel $\pi_\eta^*(\cdot)$, initial guess η_1, target credible level α, termination threshold ϵ.

Generate B bootstrap samples $\left\{\breve{\mathcal{D}}^{[b]}\right\}_{b=1}^B$ from $\mathcal{D}$.
Set $s \longleftarrow 1$ and $l \longleftarrow 1$.
`while converge`
  Compute $\hat{\boldsymbol{\theta}}$ with $\eta_{(s)}$.
  `for` $b = 1, \ldots, B$:
    Simulate R posterior draws for $\breve{\mathcal{D}}^{[b]}$ using a MCMC sampler with $\eta_{(s)}$.
    Compute the credible set $C_\alpha^{\eta_{(s)}}\left(\breve{\mathcal{D}}^{[b]}\right)$.
  `end for`
  Compute the coverage probability $\hat{c}_\alpha\left(\eta_{(s)}|\mathbb{P}_N\right)$.
  `if` $\left|\hat{c}_\alpha\left(\eta_{(s)}|\mathbb{P}_N\right) - (1-\alpha)\right| < \epsilon$:
    Set $\hat{\eta} \longleftarrow \eta_{(s)}$.
    `break`
  `else`
    Set a new learning rate $\eta_{(s+1)}$ according to (2).
    Set $s \longleftarrow s+1$.
  `end if`
`end while`
`return:` $\hat{\eta}$

2.2 Sequential Monte Carlo sampler

SMC samplers [5] are a class of Monte Carlo simulation algorithms which repeatedly apply importance sampling to a sequence of synthetic intermediate distributions $\{\pi_t\}_{t=0}^T$ to obtain collections of weighted particles $\left\{w_t^{[m]}, \boldsymbol{\theta}_t^{[m]}\right\}_{m=1}^M$. The initial distribution π_0 is the prior, $\pi_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$, and the terminal distribution π_T is the target distribution, that is, the posterior, $\pi_T(\boldsymbol{\theta}) = p(\mathcal{D}|\boldsymbol{\theta})\,p(\boldsymbol{\theta})$.

An intermediate distribution is specified as a likelihood-tempered posterior:

$$\pi_t(\boldsymbol{\theta}) \propto p(\mathcal{D}|\boldsymbol{\theta})^{\phi_t}\,p(\boldsymbol{\theta}), \tag{3}$$

where $\{\phi_t\}_{t=1}^T$ denotes an increasing sequence with $\phi_0 = 0$ and $\phi_T = 1$. ϕ_t can be interpreted as inverse temperature. $p\ \{\tilde{\pi}_t\}_{t=0}^T$ denotes a sequence of auxiliary distributions:

$$\tilde{\pi}_t(\boldsymbol{\theta}_{0:t}) = \pi_t(\boldsymbol{\theta}_t)\prod_{s=0}^{t-1}\mathcal{L}_s(\boldsymbol{\theta}_{s+1}, \boldsymbol{\theta}_s),$$

where $\mathcal{L}_t(\cdot,\cdot)$ denotes a Markov kernel which is also called a backward kernel, where it moves back from $\boldsymbol{\theta}_{t+1}$ to $\boldsymbol{\theta}_t$. $\tilde{\pi}_t$ is approximated using a system of weighted particles, $\left\{w_t^{[m]}, \boldsymbol{\theta}_t^{[m]}\right\}_{m=1}^M$. The particles are mutated via a Markov kernel $\mathcal{K}_t(\cdot,\cdot)$ which is termed a forward kernel. Let $\gamma_t(\boldsymbol{\theta})$ denote the unnormalized posterior density with ϕ_t

$$\gamma_t(\boldsymbol{\theta}) = p(\mathcal{D}|\boldsymbol{\theta})^{\phi_t}\,p(\boldsymbol{\theta}).$$

The unnormalized weights are represented as

$$\begin{aligned} W_t^{[m]} &\propto \frac{\tilde{\pi}_t\left(\boldsymbol{\theta}_t^{[m]}\right)\prod_{s=1}^{t-1}\mathcal{L}_s\left(\boldsymbol{\theta}_{s+1}^{[m]}, \boldsymbol{\theta}_s^{[m]}\right)}{\zeta_1\left(\boldsymbol{\theta}_1^{[m]}\right)\prod_{s'=1}^{t-1}\mathcal{K}_{s'}\left(\boldsymbol{\theta}_{s'-1}^{[m]}, \boldsymbol{\theta}_{s'}^{[m]}\right)} \\ &\propto \widetilde{W}_t^{[m]} W_{t-1}^{[m]}, \end{aligned}$$

where $\widetilde{W}_t^{[m]}$ represents the unnormalized incremental weight given by

$$\widetilde{W}_t^{[m]} = \frac{\gamma_t\left(\theta_t^{[m]}\right)\mathcal{L}_{t-1}\left(\theta_t^{[m]}, \theta_{t-1}^{[m]}\right)}{\gamma_{t-1}\left(\theta_{t-1}^{[m]}\right)\mathcal{K}_t\left(\theta_{t-1}^{[m]}, \theta_t^{[m]}\right)}.$$

Following the literature, we employ an MCMC kernel for the forward kernel $\mathcal{K}_t$, leaving it π_t-invariant. This choice is optimal, as the variance of the unnormalized weights is approximately minimized [5]. The backward kernel is represented as

$$\mathcal{L}_{t-1}\left(\theta_t^{[m]}, \theta_{t-1}^{[m]}\right) = \frac{\pi_t\left(\theta_{t-1}^{[m]}\right)\mathcal{K}_t\left(\theta_{t-1}^{[m]}, \theta_t^{[m]}\right)}{\pi_t\left(\theta_t^{[m]}\right)}.$$

Under this specification, the unnormalized incremental weights are reduced to

$$\widetilde{W}_t^{[m]} = \frac{\gamma_t\left(\theta_{t-1}^{[m]}\right)}{\gamma_{t-1}\left(\theta_{t-1}^{[m]}\right)} = p\left(\mathcal{D}|\theta_{t-1}^{[m]}\right)^{\phi_t-\phi_{t-1}},$$

and thus the unnormalized weights are updated as

$$W_t^{[m]} = w_{t-1}^{[m]} q\left(\theta_{t-1}^{[m]}; \mathcal{D}\right)^{\phi_t-\phi_{t-1}}.$$

As π_t evolves, the variance of the weights tends to increase, inducing the weighted particles to degenerate. The standard metric of particle degeneracy is the effective sample size (ESS) [18],

$$ESS_t = \frac{1}{\sum_{m=1}^M \left(w_t^{[m]}\right)^2} = \frac{\left(\sum_{m=1}^M W_{t-1}^{[m]}\widetilde{W}_t^{[m]}\right)^2}{\sum_{m=1}^M \left(W_{t-1}^{[m]}\widetilde{W}_t^{[m]}\right)^2}.$$

This study used stratified resampling algorithm [17] if ESS_t is below a prespecified threshold $\overline{ESS} = \psi M$, $\psi \in (0, 1)$.

The adaptive SMC sampler is applicable to generalized posterior. Exploiting the similarity between pseudo-posterior (1) in generalized/Gibbs posterior inference and likelihood tempered posterior (3) in SMC sampling, we treat the learning rate η_t analogously to the inverse temperature ϕ_t as

$$\gamma_t^*(\theta) = q(\theta; \mathcal{D})^{\eta_t} p(\theta).$$

The temperature schedule is critically important in the SMC sampler. If the temperature schedule is too coarse, the consecutive intermediate distributions are too different to obtain a good particle approximation, while if it is too fine, it incurs unnecessary computational load. In this study, we use the adaptive SMC sampler [2, 12, 13, 32]. η_{t+1} is chosen by keeping the ESS above a target level, $\widetilde{ESS}_t = \xi ESS_{t-1}$, $\xi \in (0, 1)$. The ESS can be represented as a function of η_t as follows:

$$ESS(\eta_t) = \frac{\left(\sum_{m=1}^M W_{t-1}^{[m]} q\left(\theta_{t-1}^{[m]}; \mathcal{D}\right)^{\eta_t-\eta_{t-1}}\right)^2}{\sum_{m'=1}^M \left(W_{t-1}^{[m']} q\left(\theta_{t-1}^{[m']}; \mathcal{D}\right)\right)^2}.$$

ϕ_t is chosen by solving $ESS(\eta_t) = \widetilde{ESS}_t$ via bisection search. The choice of ξ involves a trade-off between sampling quality and computational cost. The weighted particles are resampled if $ESS(\phi_1) < \overline{ESS}$. Figure 2 summarizes the adaptive SMC sampler for the generalized posterior. Let $\mathrm{ASMC}^*\left(\eta_0, \eta_T; \left\{w_0^{[m]}, \theta_0^{[m]}\right\}_{m=1}^M\right)$ denote the adaptive SMC sampler for the generalized posterior with initial and terminal learning rates η_0, η_T, respectively, and initial weighted particles $\left\{w_0^{[m]}, \theta_0^{[m]}\right\}_{m=1}^M$.

Figure 2: Adaptive SMC sampler for generalized/Gibbs posterior

`input:` dataset $\mathcal{D}$, target kernel $\pi_\eta^*(\cdot)$, initial and terminal learning rate η_1, η, target ESS $\widetilde{ESS}$, resampling threshold $\overline{ESS}$, initial weighted particles $\left\{w_0^{[m]}, \theta_0^{[m]}\right\}_{m=1}^M$.

Set $t \leftarrow 0$.
`while converge:`
  Set $t \leftarrow t+1$
  `if` $ESS(\eta) > \widetilde{ESS}$ `then:`
    Set $T \leftarrow t$ and $\eta_T \leftarrow \eta$ and
  `else:`
    Solve $ESS(\eta_t) = \widetilde{ESS}$ in $\eta_t \in (\eta_{t-1}, \eta]$.
  `end if`
  `for` $m = 1, \dots, M$`:`
    Compute the unnormalized weights
$$W_t^{[m]} = w_{t-1}^{[m]} q\left(\theta_{t-1}^{[m]}; \mathcal{D}\right)^{\eta_t-\eta_{t-1}}.$$
    Normalize the weights
$$w_t^{[m]} = W_t^{[m]}\left(\sum_{m'=1}^M W_t^{[m']}\right)^{-1}.$$
  `end for`
  Resample the particles if $ESS_t < \overline{ESS}$.
  `for` $m = 1, \dots, M$`:`
    Sample the particles $\theta_t^{[m]} \sim \mathcal{K}_t\left(\theta_{t-1}^{[m]}, \cdot\right)$.
  `end for`
  `if` $\eta_t = \eta$ `then:`
    `break`
  `end if`
`end while`
`return` $\left\{w_T^{[m]}, \theta_T^{[m]}\right\}_{m=1}^M$

2.3 Generalized posterior calibration via sequential Monte Carlo sampler

This paper proposes the GPC via SMC sampler (GPC-SMC) (Figure 3), which, as its name suggests, is a fusion of the two algorithms. Instead of an MCMC sampler, we apply the adaptive SMC sampler to bootstrap samples. The weighted particles are re-used as the initial states for the next iteration. At the sth iteration, the adaptive SMC sampler runs with a temperature schedule $\left\{\eta_{(s)}, \dots, \eta_{(s+1)}\right\}$ and initial weighted particles $\left\{w_{(s)}^{[b,m]}, \theta_{(s)}^{[b,m]}\right\}_{m=1}^M$. As the iterations proceed, a temperature schedule is likely to become shorter, drastically reducing the computational cost per iteration. In contrast, the GPC-MCMC has a constant computational cost per iteration;

Figure 3: Generalized posterior calibration via SMC sampler

`input:` observed dataset $\mathcal{D}$, target kernel $\pi_\eta^*(\cdot)$, initial guess $\eta_{(1)}$, target credible level α, termination threshold ϵ.

Set $s \leftarrow 1$ and $l \leftarrow 1$.
Compute the posterior estimate $\hat{\boldsymbol{\theta}}$ with $\eta_{(1)}$.
Generate B bootstrap samples $\left\{\breve{\mathcal{D}}^{[b]}\right\}_{b=1}^{B}$ from $\mathcal{D}$.
`for` $b = 1, ..., B$:
  Simulate M posterior draws $\left\{\boldsymbol{\theta}_{(1)}^{[b,m]}\right\}_{m=1}^{M}$ using a MCMC sampler with $\eta_{(1)}$.
  Initialize the weights $\left\{w_{(1)}^{[b,m]}\right\}_{m=1}^{M}$.
  Compute the credible set $C_\alpha^{\eta_{(1)}}\left(\breve{\mathcal{D}}^{[b]}\right)$.
`end for`
`while converge:`
  `if` $\left|\hat{c}_\alpha\left(\eta_{(s)}|\mathbb{P}_N\right) - (1-\alpha)\right| < \epsilon$ `then:`
    Set $\hat{\eta} \leftarrow \eta_{(s)}$.
    `break`
  `else:`
    Set a new learning rate $\eta_{(s+1)}$ according to (2).
    Compute the posterior estimate $\hat{\boldsymbol{\theta}}$with $\eta_{(s+1)}$.
    `for` $b = 1, ..., B$:
      Simulate the posterior draws $\left\{w_{(s+1)}^{[b,m]}, \boldsymbol{\theta}_{(s+1)}^{[b,m]}\right\}_{m=1}^{M}$ using ASMC* $\left(\eta_{(s)}, \eta_{(s+1)}; \left\{w_{(s)}^{[b,m]}, \boldsymbol{\theta}_{(s)}^{[b,m]}\right\}_{m=1}^{M}\right)$.
      Compute the credible set $C_\alpha^{\eta_{(s+1)}}\left(\breve{\mathcal{D}}^{[b]}\right)$.
    `end for`
    Set $s \leftarrow s+1$.
  `end if`
`end while`
`return:` $\hat{\eta}$

even when $\eta_{(s)}$ and $\eta_{(s+1)}$ are close during the last phase of the optimization, it needs to simulate the full length of chains.

Though we can initialize particles using an SMC sampler with a long temperature schedule $\left\{0, ..., \eta_{(1)}\right\}$, we suggest using an MCMC sampler with $\eta_{(1)}$ to generate initial particles. The corresponding weights are set according to the posterior densities evaluated at the particles.

$$\begin{aligned} W_{(1)}^{[b,m]} &= q\left(\boldsymbol{\theta}^{[b,m]}; \breve{\mathcal{D}}^{[b]}\right)^{\eta_{(1)}} p\left(\boldsymbol{\theta}^{[b,m]}\right), \\ w_{(1)}^{[b,m]} &= W_{(1)}^{[b,m]}\left(\sum_{m'=1}^{M} W_{(1)}^{[b,m']}\right)^{-1}. \end{aligned}$$

In the GPC-SMC, a temperature schedule can be decreasing if $\eta_{(s)} < \eta_{(s+1)}$. This might seem peculiar, but the SMC sampler effectively works as long as the temperature schedule is selected finely enough, that is, ξ is close to one, making the proposal and target distributions sufficiently similar.

In our context, each of an MCMC sampler and SMC sampler has its advantages and disadvantages. The computational cost of an SMC sampler is generally larger than that of a MCMC sampler. Therefore, if an efficient MCMC sampler is available, there may be no strong reason to use an SMC sampler. On the contrary, an SMC sampler works well for distributions that are difficult to simulate, e.g., posterior distributions with multimodality or discontinuity. The advantage of an SMC sampler is more pronounced for Gibbs posterior inference because no closed-form expression of the conditional posterior distributions is available, making posterior simulations difficult, and because for many cases a Gibbs posterior is discontinuous as in the subsequent section. While Hamiltonian Monte Carlo is a state-of-the-art solution to sampling from arbitrary distributions, it is only applicable to continuous target distributions. Thus, as seen below, there is no feasible option for posterior simulation except classical and sub-optimal algorithms such as the random-walk Metropolis-Hastings algorithm. Indeed, the choice of an MCMC kernel $\mathcal{K}_t$ is also important in SMC sampling, but it is not as sensitive an issue as in MCMC sampling.

3 APPLICATION

3.1 Quantile regression

For demonstration, we applied the GPC-SMC and GPC-MCMC to quantile regression with synthetic data. Following Section 4 of [27], the DGP is specified as

$$y_i = \boldsymbol{\theta}^\top \boldsymbol{x}_i + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}\left(0, \sigma^2\right),$$

where $\boldsymbol{x}_i = \left(1, x_{1,i}\right)^\top$, $x_{1,i} + 2 \sim \chi_4^2$, $\varepsilon_i \sim \mathcal{N}(0,1)$, $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top = (2,1)^\top$, and $\sigma^2 = 1$. We consider three cases of the sample size $N \in \{100, 400, 1600\}$. We infer $\boldsymbol{\theta}$ using a Gibbs posterior approach. $\boldsymbol{\theta}$ is inferred for the $(100 \times \tau)$-th percentile based on an empirical risk function given by

$$r(\boldsymbol{\theta}; \mathcal{D}) = \frac{1}{N}\sum_{i=1}^{N}\left|\left(y_i - \boldsymbol{\theta}^\top \boldsymbol{x}_i\right)\left(\tau - \mathbb{I}\left\{y_i < \boldsymbol{\theta}^\top \boldsymbol{x}_i\right\}\right)\right|.$$

We fix $\tau = 0.5$. The prior for $\boldsymbol{\theta}$ is a normal distribution with mean zero and variance ς^2, $\boldsymbol{\theta} \sim \mathcal{N}\left(\boldsymbol{0}_2, \varsigma^2 \boldsymbol{I}_2\right)$. We choose a fairly diffuse prior, $\varsigma^2 = 100^2$.

For the GPC-SMC, we employed the adaptive random-walk Metropolis-Hastings (RWMH) algorithm [9] as the MCMC kernel. At the tth iteration of SMC sampling, a proposal $\boldsymbol{\theta}'$ is generated from a multivariate normal distribution, $\boldsymbol{\theta}' \sim \mathcal{N}\left(\boldsymbol{\theta}_t^{[m]}, \zeta_t \Sigma_t\right)$, where $\zeta_t\ (>0)$ denotes a scaling parameter and Σ_{t-1} represents a covariance matrix. $\boldsymbol{\theta}'$ is accepted with probability:

$$\alpha\left(\boldsymbol{\theta}', \boldsymbol{\theta}_{t-1}^{[m]}\right) = \frac{\gamma_t^*\left(\boldsymbol{\theta}'\right)}{\gamma_t^*\left(\boldsymbol{\theta}_{t-1}^{[m]}\right)}.$$

Σ_t is chosen based on the covariance matrix of the current particles as

$$\Sigma_{t+1} \leftarrow \frac{\eta_t}{\eta_{t+1}}\frac{1}{M}\sum_{m=1}^{M} w_t^{[m]}\boldsymbol{\theta}_t^{[m]}\left(\boldsymbol{\theta}_t^{[m]}\right)^\top.$$

ζ_t is adaptively tuned on the fly according to the following updating rule:

$$\log\zeta_{t+1} \leftarrow \log\zeta_t + \varphi_t\left(\varrho^* - \varrho_t\right),$$

where ϱ_t represents the average acceptance rate at the tth iteration, ϱ^* denotes a prespecified target acceptance rate, and $\{\varphi_t\}$ indicates a decreasing sequence such that $\sum_t \varphi_t = \infty$ and $\sum_t \varphi_t^2 < \infty$. We fixed $\varrho^* = 0.25$ and $\varphi_t = (t+1)^{-0.51}$.

We evaluated the numerical efficiency of the two algorithms based on wall clock computation time. It is impossible to exactly compare the two algorithms, because the relative performance critically depends on the implementation details, in particular, the number of MCMC draws R and number of particles M. We choose $M = 1,000$ and $R = 20,000$ so that the multivariate effective sample size (multiESS) [29] is approximately the same as M. Although multiESS for MCMC sampling, which reflects the degrees of autocorrelations in the generated chain, is conceptually different from ESS for SMC sampling, they have similar implications for practitioners. As a metric of the efficiency of an MCMC sampler, a minimum of chain-wise effective sample sizes (minESS) [6] is also used in the literature. As minESS is generally smaller than multiESS (roughly a half in our case), using multiESS means that the comparison is more favorable to the GPC-MCMC than using minESS.

The other tuning parameters are selected as follows. The target credible level is $\alpha = 0.05$, meaning a 95% credible set is calibrated. The number of bootstrap samples is $B = 500$. The stopping criterion for finding the appropriate η is $\epsilon = 0.005$. The initial guess of the learning rate is fixed to $\eta = 1$. The threshold for resampling is $\psi = 0.5$, following the standard practice in the literature. The tuning parameter of the target ESS for choosing a new learning rate is $\xi = 0.999$. While there is neither consensus nor a principled guideline regarding the choice of ξ, our choice may be one of the largest values among those used in the literature [3].Thus, our choice is rather unfavorable to the GPC-SMC in comparison to the GPC-MCMC.

All the computations are conducted using MATLAB (R2023b) on an Ubuntu desktop (22.04.4 LTS) running on an AMD Ryzen Threadripper 3990X 2.9GHz 64-core processor. The computation using different bootstrap samples is parallelized. If a multi-machine parallel system is available, the GPC-SMC would run even faster, as the particle-wise computation in an SMC sampler is also parallelizable. For the GPC-MCMC, we use an algorithm proposed by Vihola [30] which is a variant of Haario et al. [9], because the adaptive RWMH algorithm does not work well, resulting in strongly autocorrelated chains. Vihola's [30] adaptive algorithm is designed to estimate the shape of the target distribution while coercing the acceptance rate. Therefore, again, our comparison is unfavorable to the GPC-SMC.

Table 1 reports the coverage probabilities based on 200 synthetic datasets. For all three cases, the coverage probabilities are close to 95%, which implies that both algorithms effectively achieved the target credible/confidence level. The medians of the calibrated learning rate are $\eta \approx 1.6$ for $N = 100$, $\eta \approx 1.4$ for $N = 400$, and $\eta \approx 1.1$ for $N = 1600$.

Figure 4 displays the distributions of the computation time. The GPC-SMC is generally faster than the GPC-MCMC. When the initial value $\eta = 1$ is close to the desirable value, the difference in computation time between the two algorithms is pronounced. This happens because the advantage of the GPC-SMC is significant for the final phase of the stochastic approximation optimization. For cases with $N = 100$, $\eta = 1$ is too small and during the initial phase, long temperature schedules are used for SMC sampling, incurring non-negligible computational cost. Therefore, with a good guess of η, the GPC-SMC would further overwhelm the GPC-MCMC.

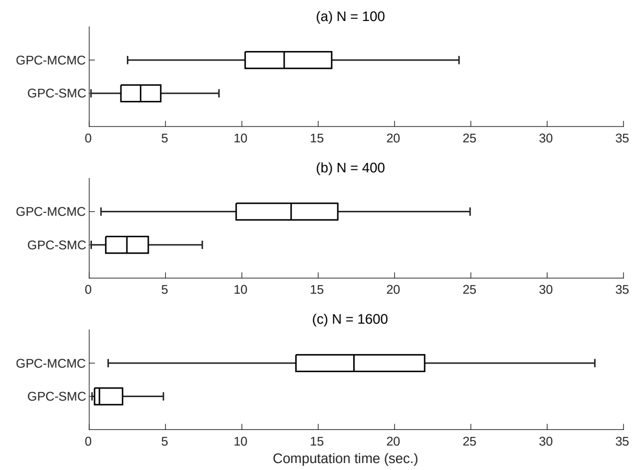
[3]While Jasra et al. [13] uses $\xi = 0.95$, there is considerable latitude in setting ξ: from 1/3 [15] to 0.999 [36].

Table 1: Coverage probability

N	Coverage probability (%)	
	GPC-MCMC	GPC-SMC
100	97.0	96.0
400	93.5	94.0
1600	95.0	94.5

Note: R is the number of draws generated by an MCMC sampler. M is the number of particles generated by an SMC sampler. Coverage probability of ground truth is evaluated based on 200 synthetic datasets.

Figure 4: Comparison of computational cost



Note: Boxplot displays the distribution of computation time (in second) for 200 synthetic datasets.

3.2 Support vector machine

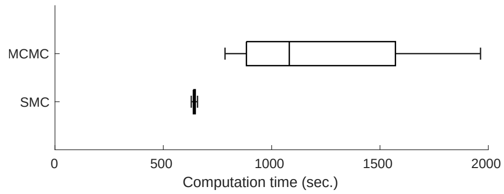
We consider support vector machine classification with the South African Heart Disease dataset (see Section 4.4.2 of [10]), as in Section 5 of [27]. We have a binary outcome $y_i \in \{-1, 1\}$ and a K-dimensional vector of predictors $\boldsymbol{x}_i = \left(1, x_{1,i}, ..., x_{K-1,i}\right)^\top$. Stacking them yields $\boldsymbol{y} = (y_1, ..., y_N)^\top$ and $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)^\top$. The support vector machine seeks to find $\boldsymbol{\theta} = (\theta_1, ..., \theta_k)^\top$ that minimizes the following objective function:

$$r(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} 2 \max\left(0, 1 - y_i \boldsymbol{\theta}^\top \boldsymbol{x}_i\right).$$

We assigned an independent Laplace-type prior to $\boldsymbol{\theta}$. The log pseudo-posterior is represented as

$$\pi_\eta^*(\boldsymbol{\theta}) \propto -\eta \sum_{i=1}^{N} 2 \max\left(0, 1 - y_i \boldsymbol{x}_i^\top \boldsymbol{\theta}\right) - \nu^{-1} \sum_{k=1}^{K} \left|\frac{\theta_k}{\sigma_k}\right|,$$

Figure 5: Comparison of computational cost



Note: Boxplot displays the distribution of computation time (in second) for 20 runs with different random seeds.

where σ_k denotes the standard deviation of the kth predictor $x_{k,1}, ..., x_{k,N}$ with $\sigma_1 = 1$ and $\nu\ (> 0)$ represents a tuning parameter. We fixed $\nu = 10$. [25] developed a Gibbs sampler for Bayesian inference of this model based on a data augmentation technique, although a resulting credible set ($\eta = 1$) is not well-calibrated [27]. We use the Gibbs sampler of [25] for the GPC-MCMC, while the adaptive RWMH for the GPC-SMC. The sample size is $N = 462$ and the dimension of $\boldsymbol{\theta}$ is $K = 8$. We choose $R = 20,000$ and $M = 4,000$ so that the multiESS from MCMC sampling and ESS from SMC sampling are roughly matched. The other tuning parameters are the same as in Section 3.1.

We executed the two algorithms 20 times with different random seeds. Both algorithms consistently yielded $\eta \approx 0.09$, which aligns with the result of [27]; the Gibbs sampling algorithm of [25] without calibration of η brings excessively optimistic uncertainty quantification. As shown in Figure 5, the GPC-SMC outperforms the GPC-MCMC in terms of computing speed. The medians of computation time for the GPC-MCMC and GPC-SMC are approximately 18.0 and 10.7 minutes, respectively. In addition, the variation in time taken for computation is much smaller for the GPC-SMC; the coefficients of variation for the GPC-MCMC and GPC-SMC are 0.335 and 0.017, respectively.

4 CONCLUSION

In conclusion, this paper presented a novel and computationally efficient strategy for selecting an appropriate learning rate in generalized posterior inference. By building upon the GPC algorithm [27], which aims to achieve nominal frequentist coverage, we devised an algorithm that combined elements of the GPC with the SMC sampler. This integration harnessed the similarity between the learning rate in generalized posterior inference and the inverse temperature in SMC sampling, enabling the calibration of the posterior distribution with reduced computational costs. The proposed approach addresses the limitation of high computational costs associated with existing methods by leveraging the efficiency of SMC sampling. Through empirical demonstrations on statistical learning models, we illustrated the effectiveness and practicality of our proposed algorithm in selecting an appropriate learning rate for robustness toward model misspecification.

Overall, this research contributes to advancing the methodology of generalized posterior inference by providing a scalable and efficient solution for learning rate selection. Future work may explore further refinements and extensions of the proposed algorithm, as well as its application in other domains requiring robust Bayesian inference under model uncertainty.

REFERENCES

[1] Yves A. Atchadé. 2017. On the Contraction Properties of Some High-dimensional Quasi-posterior Distributions. *Annals of Statistics* 45, 5 (2017), 2248–2273. https://doi.org/10.1214/16-AOS1526

[2] Alexandros Beskos, Ajay Jasra, Nikolas Kantas, and Alexandre Thiery. 2016. On the Convergence of Adaptive Sequential Monte Carlo Methods. *Annals of Applied Probability* 26, 2 (2016), 1111–1146. https://doi.org/10.1214/15-AAP1113

[3] P. G. Bissiri, C. C. Holmes, and S. G. Walker. 2016. A General Framework for Updating Belief Distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 78, 5 (02 2016), 1103–1130. https://doi.org/10.1111/rssb.12158

[4] Chenguang Dai, Pierre E. Jacob Jeremy Heng, and Nick Whiteley. 2022. An Invitation to Sequential Monte Carlo Samplers. *J. Amer. Statist. Assoc.* 117, 539 (2022), 1587–1600. https://doi.org/10.1080/01621459.2022.2087659 arXiv:https://doi.org/10.1080/01621459.2022.2087659

[5] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. 2006. Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 68, 3 (05 2006), 411–436. https://doi.org/10.1111/j.1467-9868.2006.00553.x

[6] Charles J. Geyer. 1992. Practical Markov Chain Monte Carlo. *Statist. Sci.* 7, 4 (1992), 473 – 483. https://doi.org/10.1214/ss/1177011137

[7] Peter Grünwald. 2012. The Safe Bayesian. In *Algorithmic Learning Theory*, Nader H. Bshouty, Gilles Stoltz, Nicolas Vayatis, and Thomas Zeugmann (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 169–183.

[8] Peter Grünwald and Thijs van Ommen. 2017. Inconsistency of Bayesian Inference for Misspecified Linear Models, and a Proposal for Repairing It. *Bayesian Analysis* 12, 4 (2017), 1069–1103. https://doi.org/10.1214/17-BA1085

[9] Heikki Haario, Eero Saksman, and Johanna Tamminen. 2001. An Adaptive Metropolis Algorithm. *Bernoulli* 7, 2 (2001), 223–242. https://doi.org/10.2307/3318737

[10] Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, and Jerome H. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Vol. 2. Springer.

[11] C. C. Holmes and S. G. Walker. 2017. Assigning a Value to a Power Likelihood in a General Bayesian Model. *Biometrika* 104, 2 (03 2017), 497–503. https://doi.org/10.1093/biomet/asx010

[12] Jonathan H. Huggins and Daniel M. Roy. 2019. Sequential Monte Carlo As Approximate Sampling: Bounds, Adaptive Resampling via ∞-ESS, and an Application to Particle Gibbs. *Bernoulli* 25, 1 (2019), 584–622. https://doi.org/10.3150/17-BEJ999

[13] Ajay Jasra, David A. Stephens, Arnaud Doucet, and Theodoros Tsagaris. 2011. Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo. *Scandinavian Journal of Statistics* 38, 1 (2011), 1–22. https://doi.org/10.1111/j.1467-9469.2010.00723.x

[14] Wenxin Jiang and Martin A. Tanner. 2008. Gibbs Posterior for Variable Selection in High-dimensional Classification and Data Mining. *Annals of Statistics* 36, 5 (2008), 2207–2231. https://doi.org/10.1214/07-AOS547

[15] Nikolas Kantas, Alexandros Beskos, and Ajay Jasra. 2014. Sequential Monte Carlo Methods for High-dimensional Inverse Problems: A Case Study for the Navier–Stokes Equations. *SIAM/ASA Journal on Uncertainty Quantification* 2, 1 (2014), 464–489. https://doi.org/10.1137/130930364

[16] Harry Kesten. 1958. Accelerated Stochastic Approximation. *Annals of Mathematical Statistics* 29, 1 (1958), 41–59.

[17] Genshiro Kitagawa. 1996. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics* 5, 1 (1996), 1–25. https://doi.org/10.1080/10618600.1996.10474692

[18] Augustine Kong, Jun S. Liu, and Wing Hung Wong. 1994. Sequential Imputations and Bayesian Missing Data Problems. *J. Amer. Statist. Assoc.* 89, 425 (1994), 278–288. https://doi.org/10.1080/01621459.1994.10476469

[19] Jun S. Liu. 2001. *Monte Carlo Strategies in Scientific Computing*. Springer.

[20] S. P. Lyddon, C. C. Holmes, and S. G. Walker. 2019. General Bayesian Updating and the Loss-likelihood Bootstrap. *Biometrika* 106, 2 (03 2019), 465–478. https://doi.org/10.1093/biomet/asz006

[21] Ryan Martin and Nicholas Syring. 2022. Direct Gibbs Posterior Inference on Risk Minimizers: Construction, Concentration, and Calibration. In *Advancements in Bayesian Methods and Implementation*, Arni S.R. Srinivasa Rao, G. Alastair Young, and C.R. Rao (Eds.). Handbook of Statistics, Vol. 47. Elsevier, Chapter 1, 1–41. https://doi.org/10.1016/bs.host.2022.06.004

[22] Jeffrey W. Miller. 2021. Asymptotic Normality, Concentration, and Coverage of Generalized Posteriors. *Journal of Machine Learning Research* 22, 168 (2021), 1–53.

[23] Jeffrey W. Miller and David B. Dunson. 2019. Robust Bayesian Inference via Coarsening. *J. Amer. Statist. Assoc.* 114, 527 (2019), 1113–1125. https://doi.org/10.1080/01621459.2018.1469995 PMID: 31942084.

[24] David J. Nott, Christopher Drovandi, and David T. Frazier. forthcoming. Bayesian Inference for Misspecified Generative Models. *Annual Review of Statistics and Its Application* (forthcoming). https://doi.org/10.1146/annurev-statistics-040522-015915

[25] Nicholas G. Polson and Steven L. Scott. 2011. Data Augmentation for Support Vector Machines. *Bayesian Analysis* 6, 1 (2011), 43–48.

[26] Herbert Robbins and Sutton Monro. 1951. A Stochastic Approximation Method. *Annals of Mathematical Statistics* 22, 3 (1951), 400–407. http://www.jstor.org/stable/2236626

[27] Nicholas Syring and Ryan Martin. 2019. Calibrating General Posterior Credible Regions. *Biometrika* 106, 2 (12 2019), 479–486. https://doi.org/10.1093/biomet/asy054

[28] Nicholas Syring and Ryan Martin. 2023. Gibbs Posterior Concentration Rates under Sub-exponential Type Losses. *Bernoulli* 29, 2 (2023), 1080–1108. https://doi.org/10.3150/22-BEJ1491

[29] Dootika Vats, James M. Flegal, and Galin L. Jones. 2019. Multivariate Output Analysis for Markov Chain Monte Carlo. *Biometrika* 106, 2 (04 2019), 321–337. https://doi.org/10.1093/biomet/asz002

[30] Matti Vihola. 2012. Robust Adaptive Metropolis Algorithm with Coerced Acceptance Rate. *Statistics and Computing* 22 (2012), 997–1008. https://doi.org/10.1007/s11222-011-9269-5

[31] Stephen Walker and Nils Lid Hjort. 2001. On Bayesian Consistency. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 63, 4 (2001), 811–821. https://doi.org/10.1111/1467-9868.00314

[32] Nick Whiteley, Anthony Lee, and Kari Heine. 2016. On the Role of Interaction in Sequential Monte Carlo Algorithms. *Bernoulli* 22, 1 (2016), 494–529. https://doi.org/10.3150/14-BEJ666

[33] Pei-Shien Wu and Ryan Martin. 2023. A Comparison of Learning Rate Selection Methods in Generalized Bayesian Inference. *Bayesian Analysis* 18, 1 (2023), 105–132. https://doi.org/10.1214/21-BA1302

[34] Tong Zhang. 2006. From ε-entropy to KL-entropy: Analysis of Minimum Information Complexity Density Estimation. *Annals of Statistics* 34, 5 (2006), 2180–2210. https://doi.org/10.1214/009053606000000704

[35] Tong Zhang. 2006. Information-theoretic Upper and Lower Bounds for Statistical Estimation. *IEEE Transactions on Information Theory* 52, 4 (2006), 1307–1321. https://doi.org/10.1109/TIT.2005.864439

[36] Yan Zhou, Adam M. Johansen, and John A.D. Aston. 2016. Toward Automatic Model Comparison: An Adaptive Sequential Monte Carlo Approach. *Journal of Computational and Graphical Statistics* 25, 3 (2016), 701–726. https://doi.org/10.1080/10618600.2015.1060885 arXiv:https://doi.org/10.1080/10618600.2015.1060885