STATE SPACE EMULATION AND ANNEALED SEQUENTIAL MONTE CARLO FOR HIGH DIMENSIONAL OPTIMIZATION

Chencheng Cai and Rong Chen

Washington State University and Rutgers University

Supplementary Material

In this supplementary material, we provides more details on the annealed SMC algorithm as long as additional examples of emulation and their simulation results.

S1 Conditional Sampling in Annealed SMC

In annealed SMC, at temperature $1/\kappa_k$, we need to estimate the proposal distribution $q_{k,t}(x_t \mid \boldsymbol{x}_{t-1}; \kappa_k) = \hat{p}_{k,t}(x_t \mid \boldsymbol{x}_{t-1})$ with the sample paths from the previous iteration $\{\boldsymbol{x}_{k-1,T}^{(j)}\}_{j=1,\ldots,m}$. Notice that, the weighted samples $\{(\boldsymbol{x}_{k-1,T}^{(j)}, w_{k-1,T}^{(j)})\}_{j=1,\ldots,m}$ follow the distribution $\pi(\boldsymbol{x}_t \mid \boldsymbol{y}_T; \kappa_{k-1})$. Therefore, estimating the proposal distribution is equivalent to estimating the conditional distribution from a sample set drawn from the joint distribution. Here we propose two methods to sample from such a conditional probability.

Parametric Approach.

For each time t, suppose $\{\Psi_{t,\theta}(\cdot)\}$ is a parametric family of distributions defined on $\mathcal{X}^{t+1}$ and indexed by θ. The joint distribution of $\boldsymbol{x}_t$ conditioned on $\boldsymbol{y}_T$ under κ_{k-1} is approximated by one of the distributions in the family. Specifically, let

$$\theta_{t,k-1}^* = \arg\max_{\theta} \prod_{i=1}^{m} w_{k-1,T}^{(i)} \log \psi_{t,\theta}(\boldsymbol{x}_{k-1,t}^{(i)}),$$

where $\psi_{t,\theta}$ is the corresponding probability density/mass function of $\Psi_{t,\theta}$. Denote the conditional probability induced from $\Psi_{t,\theta}(\boldsymbol{x}_t)$ as $\psi_{t,\theta}(x_t \mid \boldsymbol{x}_{t-1})$. The joint distribution of $\boldsymbol{x}_t \mid \boldsymbol{y}_T, \kappa_{k-1}$ is approximated by $\psi_{t,\theta_{t,k-1}^*}(\boldsymbol{x}_t)$ and the proposal distribution $q_t(x_t \mid \boldsymbol{x}_{t-1}; \kappa_k)$ is estimated by $\psi_{t,\theta_{t,k-1}^*}(x_t \mid \boldsymbol{x}_{t-1})$.

One common choice for the distribution family is the multivariate Gaussian distributions. In this case,

$$\psi_{t,\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_{0:t,0:t}}(\boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{0:t,0:t}\right).$$

The optimal parameter can be obtained by sample mean and sample variance such that

$$\boldsymbol{\mu}^*_{t,k-1} = \sum_{i=1}^{m} w^{(i)}_{k-1,T}\boldsymbol{x}^{(i)}_{k-1,t} \Bigg/ \sum_{i=1}^{m} w^{(i)}_{k-1,T} \,,$$

$$\boldsymbol{\Sigma}^*_{0:t,0:t,k-1} = \sum_{i=1}^{m} w^{(i)}_{k-1,T}\boldsymbol{x}^{(i)}_{k-1,t}\left[\boldsymbol{x}^{(i)}_{k-1,t}\right]' \Bigg/ \sum_{i=1}^{m} w^{(i)}_{k-1,T} \,.$$

Denote

$$\boldsymbol{\mu}^*_{t,k-1} = \begin{pmatrix} \boldsymbol{\mu}^*_{t-1,k-1} \\ \mu^*_{t,k-1} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Sigma}^*_{0:t,0:t,k-1} = \begin{bmatrix} \boldsymbol{\Sigma}^*_{0:t-1,0:t-1,k-1} & \boldsymbol{\Sigma}^*_{0:t-1,t,k-1} \\ \boldsymbol{\Sigma}^*_{t,0:t-1,k-1} & \Sigma^*_{t,t,k-1} \end{bmatrix}.$$

Then the induced conditional probability has the following closed-form:

$$p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_T; \kappa_{k-1}) = \mathcal{N}\left(\mu_{t|0:t-1,k-1}, \Sigma_{t|0:t-1,k-1}\right),$$

where the parameters are

$$\mu_{t|0:t-1,k-1} = \mu^*_{t,k-1} + \boldsymbol{\Sigma}^*_{t,0:t-1,k-1}\left[\boldsymbol{\Sigma}^*_{0:t-1,0:t-1,k-1}\right]^{-1}(\boldsymbol{x}_{t-1} - \boldsymbol{\mu}^*_{t-1,k-1}),$$

$$\Sigma_{t|0:t-1,k-1} = \Sigma^*_{t,t,k-1} - \boldsymbol{\Sigma}^*_{t,0:t-1,k-1}\left[\boldsymbol{\Sigma}^*_{0:t-1,0:t-1,k-1}\right]^{-1}\boldsymbol{\Sigma}^*_{0:t-1,t,k-1}.$$

The results above for multivariate Gaussian distributions can be easily extended to mixture Gaussian distributions, which can approximate most distributions well.

Nonparametric Approach.

When there is no appropriate distribution family to describe the joint distribution of $\boldsymbol{x}_{k-1,t}$, one can sample from the conditional distribution $p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_T; \kappa_{k-1})$ of $\{\boldsymbol{x}^{(j)}_{k-1,T}\}_{j=1,\dots,n}$ nonparametrically. Specifically, suppose $\mathcal{K}_{\boldsymbol{b}_1}(\cdot)$ and $\mathcal{K}_{b_2}(\cdot)$ are kernel functions for $\boldsymbol{x}_{t-1}$ and x_t, respectively, and it is easy to sample from $\mathcal{K}_{b_2}(\cdot)$. For any given $x^{(j)}_{k,t-1}$, Figure 1 depicts the nonparametric approach to draw $x^{(j)}_{k,t}$ from the conditional distribution $p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_T; \kappa_{k-1})$ when the samples $\{(\boldsymbol{x}^{(i)}_{k-1,T}, w^{(i)}_{k-1,T})\}_{i=1,\dots,m}$ properly weighted to $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T; \kappa_{k-1})$ are available.

Figure 1: Sample nonparametrically from a Empirical Conditional Distribution

For given $\boldsymbol{x}^{(j)}_{k,t-1}$,

- *draw* l *from* $\{1, \dots, m\}$ *with probabilities proportional to*

$$P(l = i) \propto w^{(i)}_{k-1,T} \mathcal{K}_{\boldsymbol{b}_1}(\boldsymbol{x}^{(i)}_{k-1,t-1} - \boldsymbol{x}^{(j)}_{k,t-1}).$$

- *draw* ε *from the density induced by* $\mathcal{K}_{b_2}(\cdot)$.
- *return* $x^{(j)}_{k,t} = x^{(l)}_{k-1,t} + \varepsilon$.

The parametric approach often requires the state space model to satisfy certain conditions. For example, when both state equations and observation equations are approximately linear and Gaussian, the multivariate Gaussian distribution family can be used to estimate the conditional distributions. The nonparametric approach can deal with general state space models. However, it often costs much more computing power than the parametric approach.

One issue for both approaches is the high dimensionality. Unless the system has a short memory, the conditional distribution at time t involves the high dimensional $\boldsymbol{x}_t$ and with potentially increasing dimension of parameter needed or the dimensions of spaces the nonparametric approach need to operate within. One solution for reducing dimension of the sampling problem is to use a low-dimensional sufficient statistics. Suppose $S(\boldsymbol{x}_{t-1})$ is a low-dimensional sufficient statistic such that $p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_T; \kappa_{k-1}) = p(x_t \mid S(\boldsymbol{x}_{t-1}), \boldsymbol{y}_T; \kappa_{k-1})$. Both parametric and nonparametric approaches can therefore be conducted on the joint distribution of $(x_t, S(\boldsymbol{x}_{t-1}))$, which is of lower dimension. In a Markovian system, $S(\boldsymbol{x}_{t-1}) = x_{t-1}$ and the problem reduces to sampling from a much simpler distribution. In an autoregressive system with lag δ, $S(\boldsymbol{x}_{t-1}) = \boldsymbol{x}_{t-\delta:t-1}$, which is a $\delta+1$-dimensional system. Note that since the estimated conditional distribution is used as a proposal distribution, it is often tolerable to use less accurate estimators for computational efficiency. Hence various approximation and dimension reduction tools can be used, including variational Bayes approximations

(Tzikas et al., 2008).

Another issue in estimating the conditional distribution from sequential Monte Carlo samples is the sample degeneracy. In SMC, degeneracy refers to the phenomenon that the number of distinct values for some states such as X_1 can be less than the number of Monte Carlo samples, if resampling steps are engaged. The degeneracy problem is crucial for both approaches in sampling from the conditional distribution. Therefore, at $\kappa > \kappa_0$, we suggest to conduct resampling only when all propagation steps are finished to prevent the samples from trapping into local maximums. When high degeneracy is persistent, we suggest to use post-MCMC steps (Gilks and Berzuini, 2001) to regenerate the samples. If the system is reversible and SMC can be implemented backward in t, alternating forward and backward sampling through the annealing iterations may also reduce the degeneracy problem as it starts with more diversified samples in each temperature iteration.

S2 Derivation on the Example of Cubic Smoothing Spline Emulation

Recall the objective function

$$L(\boldsymbol{y}_T) = \sum_{t=1}^{T}(y_t - m(t))^2 + \lambda \int \left[m''(t)\right]^2 dt. \tag{S2.1}$$

Following the notation in the main paper, we have the following recursive relationships:

$$a_{t+1} = a_t + b_t + c_t + d_{t+1}, \quad b_{t+1} = b_t + 2c_t + 3d_{t+1}, \quad c_{t+1} = c_t + 3d_{t+1},$$

with $c_1 = c_T = 0$. Furthermore, by substituting d_{t+1} with $(c_{t+1} - c_t)/3$ in the expressions of a_t and b_t, we have

$$a_{t+1} = a_t + b_t + (c_{t+1} + 2c_t)/3, \qquad b_{t+1} = b_t + c_t + c_{t+1}. \tag{S2.2}$$

We will use the recursive relationships in (S2.2) for the construction of state space emulation. With this notation, the second term in (S2.1) is

$$\lambda \int \left[m''(t)\right]^2 dt = \lambda \sum_{t=1}^{T-1} \int_t^{t+1} \left[6(s-t)d_{t+1} + 2c_t\right]^2 ds = \frac{4}{3}\lambda \sum_{t=1}^{T-1}(c_t^2 + c_t c_{t+1} + c_{t+1}^2).$$

In this case, the original optimization problem (S2.1) over all second order differentiable functions becomes minimizing

$$f(\boldsymbol{x}_T) = \sum_{t=1}^{T}(y_t - a_t)^2 + \frac{4}{3}\lambda \sum_{t=1}^{T-1}(c_t^2 + c_t c_{t+1} + c_{t+1}^2), \tag{S2.3}$$

where $\boldsymbol{x}_T = \{(a_t, b_t, c_t)\}_{t=1,\ldots,T}$ satisfies the recursive relationships (S2.2) and the boundary condition $c_1 = c_T = 0$. Note that $\boldsymbol{x}_t$ completely defines the cubic smoothing spline solution $\hat{m}(t)$.

With a positive inverted temperature κ, an emulated state space model is one such that whose likelihood of $\boldsymbol{x}_T$ conditioned on $y_1, \ldots, y_T$ is $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T) \propto e^{-\kappa f(\boldsymbol{x}_T)}$, with $f(\cdot)$ defined in (S2.3). One possible way to decompose $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T)$ into the likelihood of a state space model is the following.

$$\begin{aligned}\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T) &\propto \exp\left(-\kappa f(\boldsymbol{x}_T)\right)\\ &= \exp\left(-\kappa \sum_{t=1}^{T} (y_t - a_t)^2 - \frac{4\lambda\kappa}{3}(\sum_{t=1}^{T-1}(c_t^2 + c_t c_{t+1} + c_{t+1}^2)\right)\\ &= \left(\prod_{t=1}^{T} e^{-\kappa(y_t-a_t)^2}\right)\left(\prod_{t=2}^{T} e^{-\dfrac{2\lambda\kappa}{3(2-\sqrt{3})}(c_t+(2-\sqrt{3})c_{t-1})^2}\right), \quad \text{(S2.4)}\end{aligned}$$

where κ, the "temperature" parameter, controls the shape of distribution.

The second term of (S2.4) provides a construction of a first order vector auto-regressive process on $\{x_t = (a_t, b_t, c_t)\}_{t=1,\ldots,T}$ as the state equation

$$\begin{bmatrix} a_t \\ b_t \\ c_t \end{bmatrix} = \begin{bmatrix} 1 & 1 & \sqrt{3}/3 \\ 0 & 1 & \sqrt{3}-1 \\ 0 & 0 & -(2-\sqrt{3}) \end{bmatrix} \begin{bmatrix} a_{t-1} \\ b_{t-1} \\ c_{t-1} \end{bmatrix} + \begin{bmatrix} 1/3 \\ 1 \\ 1 \end{bmatrix} \eta_t, \qquad \text{(S2.5)}$$

with $\eta_t \sim \mathcal{N}(0, \sigma_b^2)$, $\quad \sigma_b^2 = 3(2-\sqrt{3})/(4\lambda\kappa)$. The first term of (S2.4) provides the observation equation of $y_t = a_t + \epsilon_t$, with $\varepsilon_t \sim \mathcal{N}(0, \sigma_y^2)$, $\quad \sigma_y^2 = 1/(2\kappa)$, and the initial values $a_1 \sim \mathcal{N}(y_1, \sigma_y^2)$, $b_1 \sim 1$, and $c_1 = 0$.

S3 Additional Example of Emulation

S3.1 Regularized Linear Regression

LASSO (Tibshirani, 1996) is a widely-used regularized linear regression estimation procedure that can perform variable selection and parameter estimation at the same time.

Consider the regression model

$$\boldsymbol{Y} = \sum_{j=1}^{p} \beta_j \boldsymbol{Z}_j + \boldsymbol{\eta}$$

where $\boldsymbol{Z}_1, \dots, \boldsymbol{Z}_p \in \mathbb{R}^n$ are the p covariates that are used to model the dependent variable $\boldsymbol{Y} \in \mathbb{R}^n$ and $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma_y^2 I_n)$. A LASSO estimator of $(\beta_1, \dots, \beta_p)$ is the minimizer of

$$f(\beta_1, \dots, \beta_p) = \|\boldsymbol{Y} - \beta_1 \boldsymbol{Z}_1 - \cdots - \beta_p \boldsymbol{Z}_p\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{S3.1}$$

For a fixed set of $(\beta_1, \dots, \beta_p)$, for $t = 1, \dots, p$, define the partial residual $\boldsymbol{\epsilon}_t$ as

$$\boldsymbol{\epsilon}_t = \boldsymbol{Y} - \sum_{l=1}^{t} \beta_l \boldsymbol{Z}_l \tag{S3.2}$$

and $\boldsymbol{\epsilon}_0 = \boldsymbol{Y}$.

Since

$$\|\boldsymbol{\epsilon}_t\|_2^2 = \|\boldsymbol{\epsilon}_{t-1} - \beta_t \boldsymbol{Z}_t\|_2^2 = \|\boldsymbol{\epsilon}_{t-1}\|_2^2 + \|\boldsymbol{Z}_t\|_2^2 \left(\beta_t - \frac{\boldsymbol{\epsilon}_{t-1}' \boldsymbol{Z}_t}{\|\boldsymbol{X}_t\|_2^2}\right)^2 - \frac{\left(\boldsymbol{\epsilon}_{t-1}' \boldsymbol{X}_t\right)^2}{\|\boldsymbol{Z}_j\|_2^2},$$

we have

$$\begin{aligned} f(\beta_1, \ldots, \beta_p) &= \|\boldsymbol{\epsilon}_p\|_2^2 + \lambda \sum_{t=1}^{p} |\beta_t| \\ &= \|\boldsymbol{Y}\|_2^2 + \sum_{t=1}^{p} \left\{ \|\boldsymbol{Z}_t\|_2^2 \left(\beta_t - \frac{\boldsymbol{\epsilon}_{t-1}' \boldsymbol{Z}_t}{\|\boldsymbol{Z}_t\|_2^2} \right)^2 - \frac{\left(\boldsymbol{\epsilon}_{t-1}' \boldsymbol{Z}_t\right)^2}{\|\boldsymbol{Z}_t\|_2^2} + \lambda |\beta_t| \right\}. \end{aligned} \tag{S3.3}$$

Let $x_t = \beta_t$ and $\boldsymbol{x}_t = (\beta_1, \ldots, \beta_t)$. An emulated state space model can be designed so that

$$\begin{aligned} \pi(\boldsymbol{x}_p) \propto \exp\{-\kappa f(\boldsymbol{x}_p)\} \propto \prod_{t=1}^{p} &\exp\left\{ -\kappa \|\boldsymbol{Z}_t\|_2^2 \left(x_t - \frac{\boldsymbol{\epsilon}_{t-1}' \boldsymbol{Z}_t}{\|\boldsymbol{Z}_t\|_2^2} \right)^2 \right\} \\ \times \prod_{t=1}^{p} &\exp\left\{ -\kappa\lambda |x_t| + \kappa \frac{(\boldsymbol{\epsilon}_{t-1}' \boldsymbol{Z}_t)^2}{\|\boldsymbol{Z}_t\|_2^2} \right\}. \end{aligned} \tag{S3.4}$$

The first term of (S3.4) leads to the state equation

$$p_t(x_t \mid \boldsymbol{x}_{t-1}) \propto \exp\left\{ -\kappa \|\boldsymbol{Z}_t\|_2^2 \left(x_t - \frac{\boldsymbol{\epsilon}_{t-1}' \boldsymbol{Z}_t}{\|\boldsymbol{Z}_t\|_2^2} \right)^2 \right\}, \tag{S3.5}$$

and the second term leads to the observation equation

$$g_t(w_t \mid \boldsymbol{x}_t) \propto \alpha_t \exp\{-\alpha_t w_t\}, \tag{S3.6}$$

where

$$\alpha_t = \exp\left\{ -\kappa\lambda |x_t| + \kappa \frac{(\boldsymbol{\epsilon}_{t-1}' \boldsymbol{Z}_t)^2}{\|\boldsymbol{Z}_t\|_2^2} \right\},$$

with observation $w_t = 0$ for all t.

Note that $\boldsymbol{\epsilon}_{t-1}$ is a function of $\boldsymbol{x}_{t-1}$ as defined in (S3.2) and is available at time t. The observation equation g_t and the observation value $w_t = 0$ are imposed to incorporate α_t in $\pi(\boldsymbol{x}_p)$. The emulation for LASSO can be extended to other penalized regression with different penalty terms by changing α_t accordingly.

S3.2 L1 Trend Filtering

L1 trend filtering (Kim et al., 2009) is a variation of Hodrick-Prescott filtering (Hodrick and Prescott, 1997). An $\ell 1$ trend filtering on $y_1, \ldots, y_T$ is defined to be the minimizer of the objective function

$$f(x_1, \ldots, x_T) = \sum_{t=1}^{T} (Y_t - x_t)^2 + \lambda \sum_{t=2}^{T-1} |x_{t-1} - 2x_t + x_{t+1}|. \tag{S3.7}$$

Minimizing (S3.7) tends to produce a piece-wise linear function due to the ℓ_1 penalty on second-order difference. An emulated state space model is designed to have the following Boltzmann likelihood function.

$$\pi(\boldsymbol{x}_T) \propto e^{-\kappa f(\boldsymbol{x}_T)/2} = \prod_{t=1}^{T} \exp\left\{-\frac{\kappa}{2}(y_t - x_t)^2\right\} \prod_{t=3}^{T} \exp\left\{-\frac{\kappa}{2\lambda}|x_t - (2x_{t-1} - x_{t-2})|\right\}. \tag{S3.8}$$

The first term of (S3.8) leads to the observation equation

$$y_t = x_t + \epsilon_t, \tag{S3.9}$$

where $\epsilon_t \sim \mathcal{N}(0, \sigma_y^2)$ with $\sigma_y^2 = 1/\kappa$. The second term of (S3.8) leads to the following second order auto-regressive process on the states

$$x_t = 2x_{t-1} - x_{t-2} + \eta_t, \tag{S3.10}$$

where $\eta_t \sim Laplace(0, \lambda_x)$ with $\lambda_x = 2/(\lambda\kappa)$.

S4 Additional Simulation Results

S4.1 LASSO Regression

In this simulation study, we consider the LASSO regression problem as discussed in Section S3.1. We set $n = 40$ observations, $p = 20$ covariates and $\sigma_y = 0.3$. The covariates $(Z_1, \ldots, Z_p)$ are generated from a multivariate normal distribution $\mathcal{N}(0, \Sigma)$ where all diagonal elements of Σ is 1 and all off-diagonal elements are 0.4. β's are generated i.i.d. according to Bernoulli(0.2). λ is set to 5 in the objective function (S3.1).

We start from the initial emulated model with the temperature parameter $\kappa = \kappa_0 = 0.05$. $m = 5000$ samples are drawn from the standard SMC algorithm under the target distribution (S3.4) with $\kappa_0 = 0.05$. The state equation (S3.5) is used as the proposal distribution and the weight is from the observation equation (S3.6) as a consequence. Resampling is done when the effective sample size is below $0.3m$. The sampled state paths are plotted in Figure 2. The estimated path for solving the original LASSO problem (S3.1) using the scikit-learn python package (Pedregosa et al., 2011) is treated as the benchmark.

In the subsequent annealing procedure, we use $m = 2000$ samples and set $\kappa_k = 1.5^k \kappa_0$ for $k = 1, \ldots, 30$. The proposal distribution used in the an-

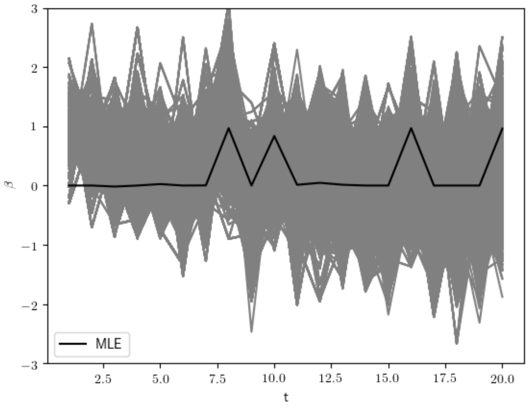


Figure 2: Sample paths at $\kappa_0 = 0.05$

nealing procedure is estimated with a multivariate normal approximation of the joint distribution of $(\beta_{k-1,t}, \ldots, \ldots, \beta_{k-1,1})$. Resampling is done only at the end of each iteration and 10 steps of post-MCMC runs are applied. The post-MCMC runs use the Gibbs sampling approach with the Metropolis-Hasting transition kernel (Metropolis et al., 1953; Hastings, 1970), where for $t = 1, \ldots, T$ and for $i = 1, \ldots, m$, a new value for β_t is proposed such that $\tilde{\beta}_t^{(i)} = \beta_t^{(i)} + \mathcal{N}(0, \tau^2)$, where $\tau^2 \propto 1/\kappa$, and the proposed move is accepted with the probability $\min(1, \pi(\tilde{\boldsymbol{x}}_t{}^{(i)} \mid \boldsymbol{y}_T; \kappa)/\pi(\boldsymbol{x}_T^{(i)} \mid \boldsymbol{y}_T; \kappa))$ with $\tilde{\boldsymbol{x}}_t^{(i)} = (\boldsymbol{x}_{t-1}^{(i)}, \tilde{x}_t^{(i)}, x_{t+1}^{(i)}, \ldots, x_T^{(i)})$. Figure 3 plots the sample paths at four different levels of κ's. Again, it is seen that the procedure is able to gradually move the sample paths towards the optimal solution. Figure 4 shows the convergence of the values of the objective function in (S3.1) evaluated at the weighted average of the sample paths.

After around 17 iterations, the weighted mean of the samples generated from the annealed SMC converges. Due to Monte Carlo variations, the sample paths and the average path cannot shrink the coefficients to exactly zero. It is tempting to run the Viterbi algorithm to refine the estimate, with zeros added to the set of allowed values of the state variables. Unfortunately the state space model designed for the LASSO problem is not Markovian hence Viterbi algorithm cannot be used. However, we used an additional refinement step by iteratively and greedily comparing each estimated state $\hat{x}_t$ (using the average sample path) with zero under the original objective function. The refinement step (with additional 0.063ms in computing time) moved some of the states to zero, and improved the value of the objective function from 21.90356 to 21.899657. The minimum achieved by the Scikit solver is 21.899645. However, such a refinement is based on the knowledge that the solution of Lasso has exactly zero coefficients, and may not be used in other optimization problems. Note that, the emulation system can be easily generalized to other types of regularization on parameters by changing the penalty term in (S3.6) without much efforts and can be adapted much more complex penalty structures.

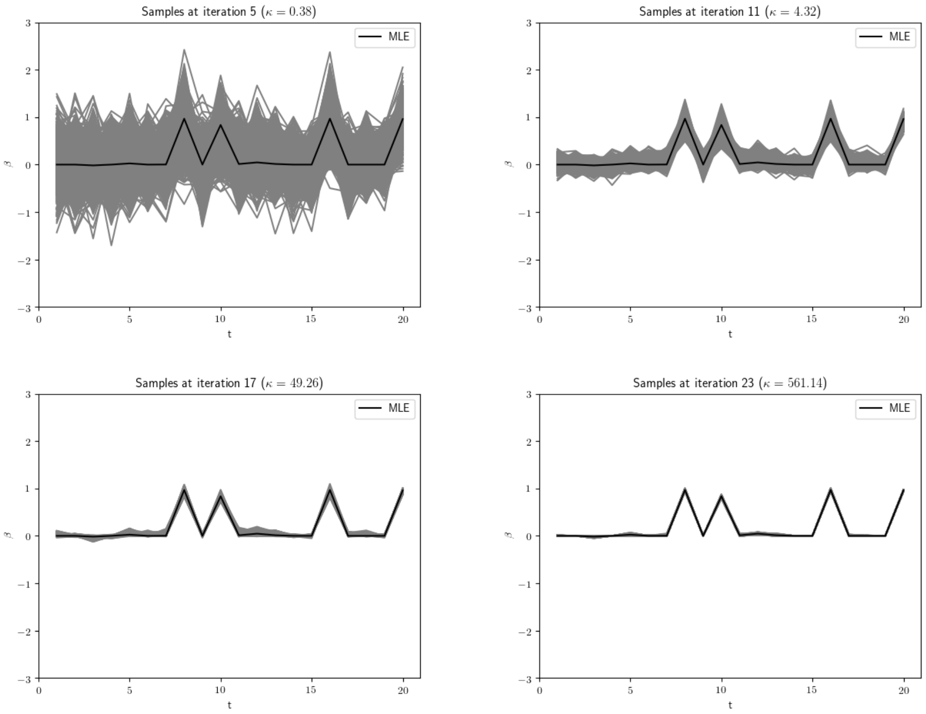


Figure 3: Sample paths at different κ's

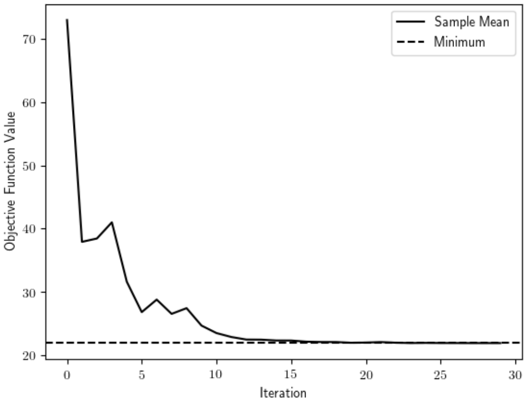


Figure 4: Value of the objective function against the number of iterations

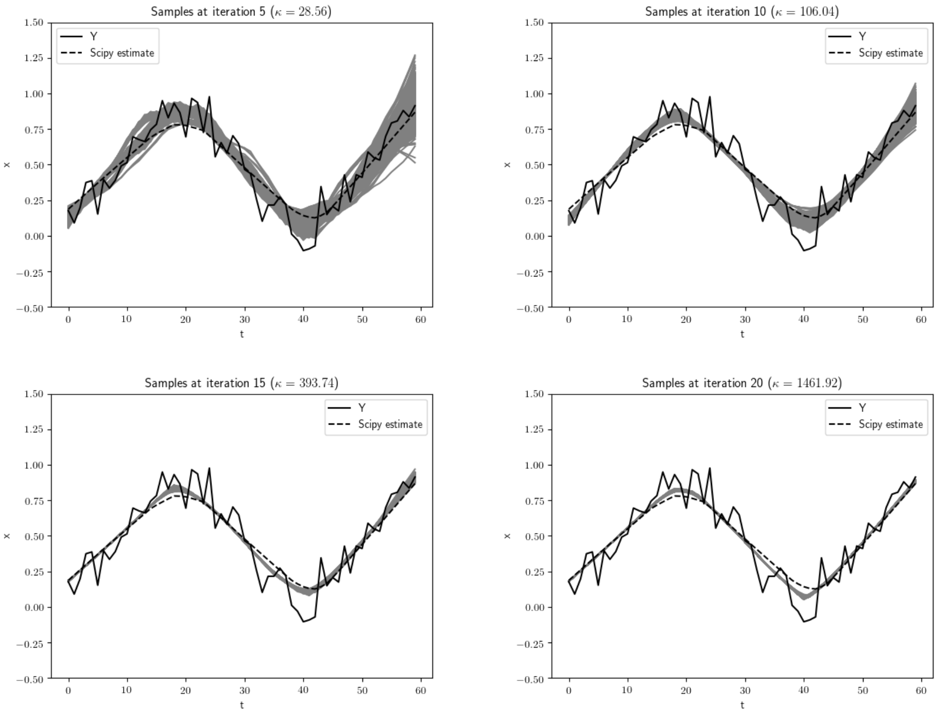
S4.2 L1 Trend Filtering

In this simulation study, we consider the ℓ_1 trend filtering problem in Section S3.2. We set $T = 60$, $\lambda = 10$ and

$$y_t = \begin{cases} \dfrac{t-1}{20} + \mathcal{N}(0, 0.01), & 1 \leqslant t \leqslant 20 \\ \dfrac{40-t}{20} + \mathcal{N}(0, 0.01), & 21 \leqslant t \leqslant 40 \\ \dfrac{t-41}{20} + \mathcal{N}(0, 0.01), & 41 \leqslant t \leqslant 60. \end{cases}$$

At $\kappa = \kappa_0 = 10$, $m = 5000$ SMC paths are sampled using the state dynamics (S3.10) as the proposal distribution. A resampling step is conducted when the effective sample size drops below $0.1m$. The approximate MLE marked as dashed line is the solution obtained by Scipy nonlinear solver. The solution shows a piece-wise linear behavior as the $\ell 1$ type of penalty appears in the objective function.

We use the following designed annealing sequence $\kappa_k = 1.3^k \kappa_0$ for $k = 1, \ldots, 40$ and use $m = 2000$ samples for annealing. In each annealing iteration, the proposal distribution used is $Laplace(\hat{E}[x_t \mid x_{t-1}, x_{t-2}; \kappa_k], \hat{V}[x_t \mid x_{t-1}, x_{t-2}; \kappa_k]^{1/2}/\sqrt{2})$ where $\hat{E}$ and $\hat{V}$ are estimated from the samples from the last iteration $\{(x_{k-1,t}^{(j)}, x_{k-1,t-1}^{(j)}, x_{k-1,t-2}^{(j)})\}_{j=1,\ldots,m}$. The Laplace distribution has a heavier tail than the normal distribution with the same variance. We found it more efficient to sample from the Laplace distribution to re-


Samples at iteration 5 ($\kappa = 28.56$)

Samples at iteration 10 ($\kappa = 106.04$)

Samples at iteration 15 ($\kappa = 393.74$)

Samples at iteration 20 ($\kappa = 1461.92$)

Figure 5: Sample paths at different κ's

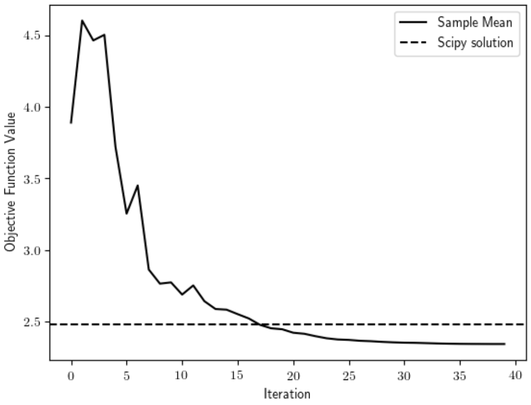


Figure 6: Value of the objective function against the number of iterations

duce sample degeneracy in this problem. The resampling step is conducted at the end of each iteration and is followed by 10 steps of post-MCMC moves. The post-MCMC steps follow the standard Gibbs sampling as in the LASSO example. Sample paths at four different κ's are displayed in Figure 5. Note that when $\kappa \approx 1462$, the sample paths are different from the nonlinear solver's solution at $t \in [38, 42]$. The value of the objective function at the sample average path shown in Figure 6 show that annealed SMC can obtained a smaller objective function value than the Scipy optimizer. The Scipy nonlinear optimizer takes 155ms while annealed SMC costs 22 ms for SMC sampling from the initial emulated model and costs around 160 ms for each subsequent annealing iteration including the post-MCMC runs.

References

Gilks, W. R. and C. Berzuini (2001). Following a moving target—monte carlo inference for dynamic bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(1), 127–146.

Hastings, W. K. (1970, 04). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57*(1), 97–109.

Hodrick, R. J. and E. C. Prescott (1997). Postwar us business cycles: an empirical investigation.

Journal of Money, credit, and Banking, 1–16.

Kim, S.-J., K. Koh, S. Boyd, and D. Gorinevsky (2009). \ell_1 trend filtering. *SIAM review 51*(2), 339–360.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics 21*(6), 1087–1092.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

Tzikas, D. G., A. C. Likas, and N. P. Galatsanos (2008). The variational approximation for bayesian inference. *IEEE Signal Processing Magazine 25*(6), 131–146.