Statistica Sinica Preprint No: SS-2022-0120	
Title	State Space Emulation and Annealed Sequential Monte Carlo for High Dimensional Optimization
Manuscript ID	SS-2022-0120
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202022.0120
Complete List of Authors	Chencheng Cai and Rong Chen
Corresponding Authors	Chencheng Cai
E-mails	chencheng.cai@temple.edu
Notice: Accepted version subject to English editing.	

Statistica Sinica

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

Chencheng Cai and Rong Chen

Temple University and Rutgers University

Abstract:

Many high dimensional optimization problems can be reformulated into the problems of finding the optimal path under an equivalent state space model setting. In this article, we present a general emulation strategy for developing a state space model whose likelihood function (or posterior distribution) shares the same general landscape as the original objective function. Then the solution of the optimization problem is the same as the optimal state path that maximizes the likelihood function. To find such an optimal path, we adapt a simulated annealing approach by inserting a temperature control into the emulated dynamic system and propose a novel annealed Sequential Monte Carlo (SMC) method that effectively generates Monte Carlo sample paths utilizing the samples obtained previously on a higher temperature scale. Compared to the vanilla simulated annealing implementation, annealed SMC is an iterative algorithm for state space model optimization that directly generates state paths from the equilibrium distributions with a decreasing sequence of temperatures through sequential importance sampling which does not require burn-in or mixing iterations to ensure quasi-equilibrium condition. Emulation examples and the corresponding simulation results are demonstrated.

Key words and phrases: Emulation, State Space Model, Sequential Monte Carlo, Optimization, Simulated Annealing

1. Introduction

High dimensional global optimization algorithms are being widely investigated since more and more applications involve high dimensional complex data nowadays. The gradient descent algorithm and its variations (Bertsekas, 1997) require the objective function to be convex or uni-modal so that the found local optimal is global. Recent research in machine learning involves many non-convex optimization problems (Anandkumar et al., 2014; Arora et al., 2012; Netrapalli et al., 2014; Agarwal et al., 2014). However, many non-convex problems remain NP-hard and the theory is only available for their convex relaxations (Jain et al., 2017). Deterministic optimization algorithms (for example, Hooke and Jeeves, 1961; Nelder and Mead, 1965; Land and Doig, 1960) may result in certain types of exhaustive search, which is computationally expensive in a high dimensional space. Stochastic optimization algorithms utilize Monte Carlo simulations to explore the parameter space in a stochastic and often more efficient way (Kiefer et al., 1952; Kirkpatrick et al., 1983; Mei et al., 2018).

In this article, we propose an emulation approach that reformulates a high dimensional optimization problem into the problem of finding the most likely state path problem in a state space model. The state space models is a class of models that describes the behavior of a usually high-dimensional random variable in a form of dynamic evolution, with wide applications in mathematics, physics and many other fields. Many high-dimensional optimization problems can be transformed to finding the optimal state path under an equivalent state space model, whose likelihood function shares the same general landscape as

the objective function of the original optimization problem. To be more specific, for a high-dimensional optimization problem with the objective function $f(x)$, we construct an emulated state space model whose likelihood function is proportional to a Boltzmann-like distribution $\exp(-\kappa f(x))$, where $\kappa > 0$ is the inverted temperature.

There are several existing heuristic approaches using the emulation idea. Cai et al. (2009) transforms a regression variable selection problem with many predictors into an optimization problem over the high dimensional binary space $\{0, 1\}^p$, which can be further converted to the most likely path problem in a state space model with binary-valued states indicating the variable selection, even though the predictors have no chronological order in nature. Kolm and Ritter (2015) reformulates a portfolio optimization problem to a state space model by mapping the utility function to the log-likelihood function. The utility function is then optimized through finding the most likely path in the corresponding state space model by applying the Viterbi algorithm (Viterbi, 1967) over Monte Carlo samples. Similarly, Irie and West (2016) relates the multi-period portfolio optimization problem to the log-likelihood of a mixture of linear Gaussian dynamic systems and proposed an algorithm based on the Kalman filter (Kalman, 1960) and EM algorithm (Dempster et al., 1977) to find the most likely path. Iglesias et al. (2013) and Zhang et al. (2021) reformulated the inversion problems to state space models by segmenting the observations into a sequence and optimizing the hidden path through Kalman filter and ensemble Kalman filter.

These studies map high dimensional optimizations to problems under state space model settings. However, it remains a challenging problem to find the most likely path analytically

and numerically. For example, the approach in Cai et al. (2009) is difficult to be generalized to continuous spaces. The Viterbi algorithm used in Kolm and Ritter (2015) requires the dynamic system to be Markovian and non-singular and it needs a large sample size in general to achieve high accuracy. The combination of Kalman filter and EM algorithm proposed in Irie and West (2016) works only when the underlying distribution can be well-represented by the mixture of Gaussian distributions.

In this paper, we propose a new Sequential Monte Carlo (SMC) based simulated annealing approach, named "annealed SMC", to find the most likely path in a state space model. The SMC algorithm is a class of Monte Carlo methods that draws samples from the state space model systems in a sequential fashion. With the sequential importance sampling and resampling (SISR) scheme, SMC is extremely powerful in sampling from complex dynamic systems, especially for the state space models (Gordon et al., 1993; Kitagawa, 1996; Kong et al., 1994; Liu and Chen, 1995, 1998; Pitt and Shephard, 1999; Chen et al., 2000; Doucet et al., 2001). Recall that the likelihood function of the emulated state space model is designed to be proportional to $\exp(-\kappa f(x))$, where κ is the inverted temperature. To mimic the (physical) annealing procedure in a non-interactive, non-quantum thermodynamic system (Kirkpatrick et al., 1983), we choose a sequence of decreasing temperatures $\kappa_0 < \kappa_1 < \cdots < \kappa_K$, which corresponds to a sequence of emulated state space models.

We start from drawing sample paths from the base emulated state space model at a high base temperature κ_0. Samples from a low temperature (large κ) system are close to the optimal sample path since the distribution is sharp at a low temperature, but drawing

from such a distribution directly is usually difficult. With annealed SMC, samples of a low temperature system can be obtained by utilizing samples obtained at a higher temperature. Eventually, all the SMC sample paths converge to the most likely one. The sequence of temperatures $\kappa_0 < \kappa_1 < \cdots < \kappa_K$ provides a slow-changing path from the base emulated state space model at κ_0, which is easy to sample from but not very useful for optimization, to the target emulated state space model at κ_K, which is difficult to sample but provides solutions to the optimization problem.

The contribution of this paper involves two main components: reformulating the problem into an emulated space space model and an annealed SMC algorithm to find the solution. In the main content, two neat examples are provided where the emulated state space models are natural, simple and illustrative. Two more examples are provided in the supplementary material to demonstrate the flexibility of the proposed method in solving existing optimization problems with some new applications.

The rest of the paper is organized as follows. Section 2 first briefly reviews state space models then introduces the principles of state space emulation. Two illustrative emulation examples are provided in Section 2.3. Section 3 introduces the framework of annealed SMC designed to find the most likely path. Simulation results corresponding to the two examples in Section 2.3 are shown in Section 4. Section 5 concludes. Two additional examples are included in the supplementary material.

2. State Space Model and State Space Emulation

2.1 State Space Model

State space model is a class of models for describing the mechanism of sequential observations $\boldsymbol{y}_T = (y_1, \ldots, y_T)$ with a sequence of latent variables $\boldsymbol{x}_T = (x_1, \ldots, x_T)$. The latent variables $\boldsymbol{x}_T$ are assumed to follow a discrete-time stochastic process governed by the state equations

$$p(x_t \mid \boldsymbol{x}_{t-1}) = p_t(x_t \mid \boldsymbol{x}_{t-1}), \tag{2.1}$$

for $t = 2, \ldots, T$, and x_1 follows its marginal distribution $p_1(x_1)$. When the distribution of x_t conditioned on $\boldsymbol{x}_{t-1}$ does not depend on $\boldsymbol{x}_{t-2}$ such that $p(x_t \mid \boldsymbol{x}_{t-1}) = p(x_t \mid x_{t-1})$, the system is Markovian. The observations $\boldsymbol{y}_T$ are generated independently condition on the latent variables through the observational equations

$$p(y_t \mid x_t) = g_t(y_t \mid x_t), \tag{2.2}$$

for $t = 1, \ldots, T$. In inference problems, the formulas of the state equations $p_t(\cdot)$ and the observation equations $g_t(\cdot)$ are usually known except for a set of unknown parameter of interest θ. In this paper, we assume $p_t(\cdot)$ and $g_t(\cdot)$ are completely known, and we are interested in inferring the latent states $\boldsymbol{x}_T$. Estimating $\boldsymbol{x}_T$ from the observations $\boldsymbol{y}_T$ under the likelihood principle is known as the most likely path (MLP) problem in hidden Markov models.

The state equations provide the prior information on $\boldsymbol{x}_T$

$$\pi(\boldsymbol{x}_T) \propto p_1(x_1) \prod_{t=2}^{T} p_t(x_t \mid \boldsymbol{x}_{t-1}), \tag{2.3}$$

and the observation equations serve as the likelihood functions

$$p(\boldsymbol{y}_T \mid \boldsymbol{x}_T) = \prod_{t=1}^{T} g_t(y_t \mid x_t). \tag{2.4}$$

A maximum-a-posterior (MAP) estimator can be obtained by maximizing the posterior function in (2.5).

$$\pi(\boldsymbol{x}_T|\boldsymbol{y}_T) \propto p_1(x_1)g_1(y_1 \mid x_1) \prod_{t=2}^{T} p_t(x_t \mid \boldsymbol{x}_{t-1})g_t(y_t \mid x_t). \tag{2.5}$$

When both $p_t(\cdot)$ and $g_t(\cdot)$ are Gaussian, the maximum of (2.5) can be obtained easily using Kalman filter and smoother (Kalman, 1960). In general cases when the analytic solution to optimize (2.5) is infeasible, the MAP estimator can be obtained by drawing sample paths $\{(x_1^{(i)}, \ldots, x_T^{(i)})\}_{i=1,\ldots,n}$ from the posterior distribution (2.5). We will discuss details in estimating most likely path using Monte Carlo methods in Section 3.

2.2 State Space Emulation

We propose a state space emulation approach for solving high dimensional optimization problems. The approach constructs a state space model so that the original optimization problem is equivalent to finding the most likely state path under the state space model.

Let $f : \mathcal{X}^d \to \mathbb{R}$ be the objective function to be minimized and $\xi : \mathbb{R} \to [0, +\infty)$ be a monotone decreasing function. Then minimizing $f(x)$ is equivalent to maximizing $\phi(x) := \xi(f(x))$ such that

$$\arg\min_{x \in \mathcal{X}^d} f(x) = \arg\max_{x \in \mathcal{X}^d} \phi(x).$$

Furthermore, if there exists a state space model whose posterior function (2.5) is proportional to $\phi(x)$ such that $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T) \propto \phi(\boldsymbol{x}_T) = \xi(f(\boldsymbol{x}_T))$, with artificially designed state equations $\{p_t(\cdot)\}_{t=1,\ldots,T}$, observation equations $\{g_t(\cdot)\}_{t=1,\ldots,T}$ and $T = d$, we call the state space model an "emulated" state space model. The observations $\boldsymbol{y}_T$ can be either certain observations involving in the original optimization problem (e.g. the observed points in the smoothing spline problem in Section 2.3.1) or artificially designed. Note that it is always possible to rewrite any joint distribution function $\phi(\boldsymbol{x}_T)$ in the form of (2.3) as $\phi(\boldsymbol{x}_T) = \phi(x_1, \ldots, x_T) = \phi_1(x_1) \prod_{t=2}^{T} \phi_t(x_t \mid \boldsymbol{x}_{t-1})$, where $\phi_t(x_t \mid \boldsymbol{x}_{t-1}) = \int_{\mathcal{X}^{T-t}} \phi(\boldsymbol{x}_T) dx_{t+1} \ldots d_T / \int_{\mathcal{X}^{T-t+1}} \phi(\boldsymbol{x}_T) dx_t \ldots d_T$ and $\phi_1(x_1) = \int_{\mathcal{X}^{t-1}} \phi(\boldsymbol{x}_T) dx_2 \ldots dx_T$. Often such a series of conditional distribution is difficult to sample from or to be evaluated.

However, in certain problems as our examples shown later, it is possible to reformulate the conditional distribution to $\phi_t(x_t \mid \boldsymbol{x}_{t-1}) = p_t(x_t \mid \boldsymbol{x}_{t-1}) g_t(y_t \mid x_t)$, in which $p_t(x_t \mid \boldsymbol{x}_{t-1})$ is easy to generate sample from and $g_t(y_t \mid x_t)$ is easy to be evaluated, for some designed y_t. In general, objective functions with local dependence between parameters can be easily emulated by Markovian state space models as in the examples of smoothing splines, trend filtering and the optimal trading path. Objective functions with more complex interactions between parameters usually lead to non-Markovian emulated state space models, which need more carefully designs. The lasso regression in the supplementary material is one such case.

Minimizing the objective function is then equivalent to finding the most likely path for the emulated state space model. The emulated state and observation equations provide guidance for further SMC implementation, even though they are artificial.

A common choice for $\xi(\cdot)$ is the Boltzmann distribution function

$$\xi(s) = e^{-\kappa s}, \tag{2.6}$$

where κ is a positive constant that relates to the temperature in statistical physics. In statistics, the Boltzmann function in (2.6) links the least square method to the maximum likelihood approach with i.i.d. Gaussian noise. With this choice of $\xi(\cdot)$, the system has a physical interpretation: The objective function $f(\cdot)$ is regarded as the possible energy levels in a non-quantum thermodynamic system. Assuming no interactions, the number of particles at the energy $f(x)$ follows the Boltzmann distribution under thermodynamic equilibrium. The integrability of $\phi(x)$ ensures the existence of the canonical partition function such that this physical canonical system is valid. The minimization of $f(\cdot)$ is now equivalent to finding the base energy level, which inspires the use of simulated annealing of this thermodynamic system. More details will be discussed in Section 3.

2.3 Examples

2.3.1 Cubic Smoothing Spline

Consider a nonparametric regression model $y_t = m(x_t) + \epsilon_t$ with equally spaced x_t. Without loss of generality, let $x_t = t$ and treat them as time. The cubic smoothing spline method (Green and Silverman, 1993) estimates a continuous function $m(t)$ by minimizing

$$L(\boldsymbol{y}_T) = \sum_{t=1}^{T} (y_t - m(t))^2 + \lambda \int \left[m''(t)\right]^2 dt. \tag{2.7}$$

The first term in (2.7) is the total squared tracking errors at the observation times and the second term is the penalty term on the smoothness of the latent function $m(\cdot)$, where λ controls the regularization strength. Given the values of $m(1), \ldots, m(T)$, the minimizer of the second term is a natural cubic spline that interpolates $m(1), \ldots, m(T)$ (see Green and Silverman (1993)). Hence, the solution to minimize (2.7) is a natural cubic spline, which is second-order continuously differentiable and is a cubic polynomial in all intervals $[t, t+1]$ for $t = 1, \ldots, T-1$ and is linear outside $[1, T]$.

Define the derivatives of $m(t)$ at each observation at time t as

$$a_t = m(t), \quad b_t = m'(t), \quad c_t = m''(t)/2, \quad d_t = \lim_{s \to t_-} m'''(s)/6.$$

The natural cubic spline solution to (2.7) is equivalent to an emulated state space model on $x_t = (a_t, b_t, c_t)$ with a vector auto-regressive state equation

$$\begin{bmatrix} a_t \\ b_t \\ c_t \end{bmatrix} = \begin{bmatrix} 1 & 1 & \sqrt{3}/3 \\ 0 & 1 & \sqrt{3}-1 \\ 0 & 0 & -(2-\sqrt{3}) \end{bmatrix} \begin{bmatrix} a_{t-1} \\ b_{t-1} \\ c_{t-1} \end{bmatrix} + \begin{bmatrix} 1/3 \\ 1 \\ 1 \end{bmatrix} \eta_t, \tag{2.8}$$

with $\eta_t \sim \mathcal{N}(0, \sigma_b^2)$, $\sigma_b^2 = 3(2-\sqrt{3})/(4\lambda\kappa)$. The corresponding observation equation is $y_t = a_t + \epsilon_t$, with $\varepsilon_t \sim \mathcal{N}(0, \sigma_y^2)$, $\sigma_y^2 = 1/(2\kappa)$, and the initial values $a_1 \sim \mathcal{N}(y_1, \sigma_y^2)$, $b_1 \sim 1$, and $c_1 = 0$. The derivation is postponed to the supplementary material.

2.3.2 Optimal Trading Path

In asset portfolio management, the optimal trading path problem is a class of optimization problems which typically maximizes certain utility functions of the trading path (Markowitz,

1959). Kolm and Ritter (2015) and Irie and West (2016) proposed to turn such an optimization problem to an emulated state space model. To be more specific, let $\boldsymbol{x}_T = (x_0, \ldots, x_T)$ be a trading path in which x_t represents the position held at time t. Kolm and Ritter (2015) propose to maximize the following utility function.

$$u(\boldsymbol{x}_T) = -\sum_{t=1}^{T} c_t(x_t - x_{t-1}) - \sum_{t=0}^{T} h_t(y_t - x_t), \tag{2.9}$$

where $(y_0, \ldots, y_T)$ is a predetermined optimal trading path in an ideal world without trading costs, typically obtained by maximizing the risk-adjusted expected return under the Markowitz mean-variance theory (Markowitz, 1959). Kolm and Ritter (2015) provides a construction of $(y_0, \ldots, y_T)$ based on the term structure of the underlying asset's *alpha* (the excess expected return relative to the market). Let $c_t(\cdot)$ represent the transaction cost which is often assumed to be a quadratic function of the absolute position change $|x_t - x_{t-1}|$. Without loss of generality, we parametrize it as

$$c_t(|x_t - x_{t-1}|) = \frac{1}{2\sigma_x^2}\left(|x_t - x_{t-1}|^2 + 2\alpha|x_t - x_{t-1}| + \alpha^2\right),$$

where α is a non-negative constant related to the volatility and liquidity of the asset (Kyle and Obizhaeva, 2011). Let $h_t(\cdot)$ be the utility loss due to the departure of the realized path from the ideal path. We use the squared loss $h_t(y_t - x_t) = (y_t - x_t)^2/(2\sigma_y^2)$. Then the objective function is

$$e^{-\kappa u(\boldsymbol{x}_T)} \propto \prod_{t=1}^{T} \exp\left(-\frac{\kappa(|x_t - x_{t-1}| + \alpha)^2}{2\sigma_x^2}\right) \prod_{t=1}^{T} \exp\left(-\frac{\kappa(y_t - x_t)^2}{2\sigma_y^2}\right).$$

Taking the position constraint $x_0 = x_T$ into consideration as discussed in Cai et al. (2018), an emulated state space model can therefore be constructed as

$$p_t(x_t \mid x_{t-1}) \propto \exp\left(-\frac{\kappa(|x_t - x_{t-1}| + \alpha)^2}{2\sigma_x^2}\right), \tag{2.10}$$

$$g_t(y_t \mid x_t) \propto \exp\left(-\frac{\kappa(y_t - x_t)^2}{2\sigma_y^2}\right). \tag{2.11}$$

With the state equation (2.10) and the observation equation (2.11), the state space model has a likelihood function proportional to $\exp(-\kappa u(\boldsymbol{x}_T))$.

3. Annealed Sequential Monte Carlo

3.1 Sequential Monte Carlo (SMC)

The sequential Monte Carlo (SMC) method is a class of sampling methods designed for state space models. It utilizes the sequential nature of state space models and draws samples incrementally with sequential importance sampling and resampling (SISR) schemes. A typical SMC approach is demonstrated in Figure 1.

The function $q_t(\cdot)$ in the propagation step in Figure 1 is the proposal distribution. As discussed in Lin et al. (2013), the "perfect" choice for the proposal is the conditional distribution with full information set such that $q_t(x_t \mid \boldsymbol{x}_{t-1}) = p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_T)$. However, in most cases, this conditional probability is impossible to evaluate or to sample from at time t. The priority score β_t is the weight used in the resampling step, which quantifies the sampler's preference over different sample paths. The most common choice of β_t is $\beta_t^{(i)} \propto w_t^{(i)}$. Different variations of the SMC algorithm choose different proposal distributions and different priority

Figure 1: Sequential Monte Carlo (SMC) Algorithm

- *Draw* $x_1^{(i)}$ *from* $p_1(x_1)$ *and set weight* $w_0^{(i)} = 1$ *for* $i = 1, \ldots, n$.
- *For time* $t = 2, \cdots, T$:
  - *Propagation: For* $i = 1, \cdots, n$,
    - *Draw* $x_t^{(i)}$ *from* $q_t(x_t \mid \boldsymbol{x}_{t-1}^{(i)})$ *and set* $\boldsymbol{x}_t^{(i)} = (x_{t-1}^{(i)}, x_t^{(i)})$.
    - *Update weights by setting*

$$w_t^{(i)} \leftarrow w_{t-1}^{(i)} \cdot \frac{p_t(x_t^{(i)} \mid \boldsymbol{x}_{t-1}^{(i)}) g_t(y_t \mid \boldsymbol{x}_t^{(i)})}{q_t(x_t^{(i)} \mid \boldsymbol{x}_{t-1}^{(i)})}.$$

  - *Resampling (optional):*
    - *Assign a priority score* $\beta_t^{(i)}$ *to each sample* $x_{0:t}^{(i)}$.
    - *Draw samples* $\{J_1, \ldots, J_n\}$ *from the set* $\{1, \ldots, n\}$ *with replacement, with probabilities proportional to* $\{\beta_t^{(i)}\}_{i=1,\ldots,n}$.
    - *Let* $\boldsymbol{x}_t^{*(i)} = \boldsymbol{x}_t^{(J_i)}$ *and* $w_t^{*(i)} = w_t^{(J_i)} / \beta_t^{(J_i)}$.
    - *Set* $\{(\boldsymbol{x}_t^{(i)}, w_t^{(i)})\}_{i=1,\ldots,n} \leftarrow \{(\boldsymbol{x}_t^{*(i)}, w_t^{*(i)})\}_{i=1,\ldots,n}$.
- *Return the weighted sample set* $\{(\boldsymbol{x}_T^{(i)}, w_T^{(i)})\}_{i=1,\ldots,n}$.

scores. The Bayesian particle filter (Gordon et al., 1993) sets $q_t(x_t \mid \boldsymbol{x}_{t-1}) = p_t(x_t \mid \boldsymbol{x}_{t-1})$. It works well when the observations y_T are relatively noisy compared with the state equation part. With accurate observations, the independent particle filter (Lin et al., 2005) uses $q_t(x_t \mid \boldsymbol{x}_{t-1}) \propto g_t(y_t \mid x_t)$. As an important (with certain additional cost) compromise over the Bayesian particle filter and the independent particle filter, Kong et al. (1994) and Liu and Chen (1998) suggests to adopt $q_t(x_t \mid \boldsymbol{x}_{t-1}) \propto p_t(x_t \mid \boldsymbol{x}_{t-1}) g_t(y_t \mid x_t)$ to reduce variance. Other sequential Monte Carlo methods focus on finding more appropriate priority scores in resampling with the help of future information. The auxiliary particle filter (Pitt and Shephard, 1999) conducts resampling with the priority score $\beta_t^{(i)} = w_t^{(i)} p(y_{t+1} \mid x_t)$. The delayed sampling method (Chen et al., 2000; Lin et al., 2013) looks ahead Δ steps further and uses $\beta_t^{(i)} = w_t^{(i)} p(y_{t+1}, \ldots, y_{t+\Delta} \mid x_t)$.

In emulations for the optimizations, we are more interested in generating samples in the high probability density region of $\pi(\boldsymbol{x}_T)$, hence our problem is essentially a smoothing problem. Briers et al. (2010) proposed to use a generalization of two-filter smoothing formula to sample approximately from the joint distribution $\pi(\boldsymbol{x}_T)$. Additional local MCMC moves can be adopted to fight degeneracy (Gilks and Berzuini, 2001). Many other SMC smoothing algorithm implementations are proposed to reduce the potential degeneracy in samples. See, for example, Godsill et al. (2004); Del Moral et al. (2010); Briers et al. (2010); Guarniero et al. (2017).

3.2 Finding the Most Likely Path

With emulation, finding the optimum of $f(\boldsymbol{x})$ is now equivalent to finding the mode, or the most likely state path (MLP), of $\pi(\boldsymbol{x}_T)$,

$$\boldsymbol{x}_T^* = \arg\max_{\boldsymbol{x}_T \in \mathcal{X}^T} \pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T), \tag{3.12}$$

with $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T)$ defined in (2.5) and $\mathcal{X}$ being the common support for all latent variables. By construction, the mode, which is the optimum of $f(\boldsymbol{x})$, does not depend on κ used in (2.6).

In this article, we focus on finding the MLP from Monte Carlo samples. A set of weighted Monte Carlo samples from the distribution $\pi(\boldsymbol{x}_T)$ can be generated by Sequential Monte Carlo (SMC) and its various implementation schemes. Let $\{(\boldsymbol{x}_T^{(i)}, w_T^{(i)})\}_{i=1,\ldots,n}$ be the samples drawn from the emulated state space model using the SMC algorithm in Figure 1. A natural and easy way is to use the empirical MAP path such that

$$\hat{\boldsymbol{x}}_T^{(map)} = \arg\max_{\boldsymbol{x}_T \in \{\boldsymbol{x}_T^{(i)}\}_{i=1,\ldots,n}} \pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T). \tag{3.13}$$

Although the empirical MAP involves the least computation given the Monte Carlo samples, it usually requires a very large sample size to achieve high accuracy, especially when the dimension T is large.

Note that the MLP is the same under different κ's. However the distribution $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T, \kappa)$ is more flat for small κ (high temperature) and is more concentrated around the MLP for large κ. Hence the empirical MAP path tends to be more accurate if the Monte Carlo

samples are generated from the target distribution with large κ. When κ is sufficiently large, the average sample path is also a good estimate of the MAP. However, it is much more difficult to generate Monte Carlo samples with large κ due to the tendency of being trapped in local optimum. Simulated annealing gradually modifies the easily generated samples at a higher temperature to obtain the samples from a lower temperature system with more accurate estimates.

3.3 Annealed SMC

We propose a simulated annealing algorithm for sequential Monte Carlo on state space models. The idea comes from the thermodynamics analogue discussed in the previous section. When the function $\xi(\cdot)$ is chosen to be Boltzmann-like as in (2.6), the Monte Carlo samples from the emulated state space model correspond to a random sample set from the non-interacting particles in a thermodynamic equilibrium system as discussed in Section 2.2.

If the temperature cools down to 0 slowly enough such that the system is approximately in thermodynamic equilibrium for any temperature in between, all particles will condense to the base energy level. The idea of simulated annealing to analogize the physical system was proposed and discussed in Kirkpatrick et al. (1983).

To mimic the thermodynamic procedure, we propose the following system to simulate the annealing procedure for the SMC samples. Let $0 < \kappa_0 < \kappa_1 < \cdots < \kappa_K$ be an increasing sequence of inverse temperatures. Suppose at κ_0, a base emulated state space model is

constructed as

$$\pi(\boldsymbol{x}_T;\kappa_0) \propto e^{-\kappa_0 f(\boldsymbol{x}_T)} \propto p_0(x_0)\prod_{t=1}^{T} p_t(x_t \mid \boldsymbol{x}_{t-1}) g_t(y_t \mid x_t). \tag{3.14}$$

At a higher inverse temperature κ_k, an emulated state space model can be induced from (3.14) such that

$$\pi(\boldsymbol{x}_T;\kappa_k) \propto e^{-\kappa_k f(\boldsymbol{x}_T)} \propto p_0(x_0;\kappa_k)\prod_{t=1}^{T} p_t(x_t \mid \boldsymbol{x}_{t-1};\kappa_k) g_t(y_t \mid x_t;\kappa_k), \tag{3.15}$$

where $p_t(x_t \mid \boldsymbol{x}_{t-1};\kappa_k) \propto [p_t(x_t \mid \boldsymbol{x}_{t-1})]^{\kappa_k/\kappa_0}$ and $g_t(y_t \mid x_t;\kappa_k) \propto [g_t(y_t \mid x_t)]^{\kappa_k/\kappa_0}$ are the corresponding state equations and observation equations at κ_k. The starting inverse temperature κ_0 is usually chosen to be relatively small such that the function $\pi(\boldsymbol{x}_T;\kappa_0) \propto e^{-\kappa_0 f(\boldsymbol{x}_T)}$ is relatively flat and is easy to sample from by SMC. We start with κ_0, draw $\{(\boldsymbol{x}_{0,T}^{(j)}, w_{0,T}^{(j)})\}_{j=1,\ldots,m}$ from the base emulated state space model $\pi(\boldsymbol{x}_T;\kappa_0)$. For $k = 1,\ldots,K$, new samples $\{(\boldsymbol{x}_{k,T}^{(j)}, w_{k,T}^{(j)})\}_{j=1,\ldots,m}$ are drawn with respect to the distribution $\pi(\boldsymbol{x}_T;\kappa_k)$ utilizing samples $\{(\boldsymbol{x}_{k-1,T}^{(j)}, w_{k-1,T}^{(j)})\}_{j=1,\ldots,m}$ obtained at κ_{k-1}. The procedure is depicted in Figure 2. The annealed sequential Monte Carlo uses the following proposal distribution at temperature κ_k:

$$q_{k,t}(x_t \mid \boldsymbol{x}_{t-1};\kappa_k) \propto \hat{p}_{k,t}(x_t \mid \boldsymbol{x}_{t-1};\kappa_{k-1}), \tag{3.16}$$

where the conditional distribution $\hat{p}_{k,t}(x_t \mid \boldsymbol{x}_{t-1};\kappa_{k-1})$ is an estimate of $\pi_T(x_t \mid \boldsymbol{x}_{t-1};\kappa_{k-1})$ and can be obtained from the Monte Carlo samples $\{(\boldsymbol{x}_{k-1,T}^{(j)}, w_{k-1,T}^{(j)})\}_{j=1,\ldots,m}$ under κ_{k-1}. We will discuss how to obtain such an estimate later. Since κ increases slowly, $\pi_T(x_t \mid \boldsymbol{x}_{t-1};\kappa_{k-1})$ and $\pi_T(x_t \mid \boldsymbol{x}_{t-1};\kappa_k)$ are reasonably close. With a sufficiently large terminating κ_K, samples

Figure 2: Annealed Sequential Monte Carlo Algorithm

- *Draw* $\{(\boldsymbol{x}_{0,T}^{(j)}, w_{0,T}^{(j)})\}_{j=1,\ldots,m}$ *from* $\pi(\boldsymbol{x}_T; \kappa_0)$ *with SMC in Figure 1, using a set of proposal distributions* $q_{1,t}(x_t \mid \boldsymbol{x}_{t-1}; \kappa_0)$.

- *For* $k = 1, \ldots, K$, *draw* $\{(\boldsymbol{x}_{k,T}^{(j)}, w_{k,T}^{(j)})\}_{j=1,\ldots,m}$ *from* $\pi(\boldsymbol{x}_T; \kappa_k)$ *with SMC in Figure 1 using the proposal distribution*

$$q_{k,t}(x_t \mid \boldsymbol{x}_{t-1}; \kappa_k) \propto \hat{p}_{k,t}(x_t \mid \boldsymbol{x}_{k,t-1}^{(j)}),$$

  where the right hand side is an estimate of $\pi_T(x_t \mid \boldsymbol{x}_{t-1}; \kappa_{k-1})$.

- *Estimate the most likely path from* $\{(\boldsymbol{x}_{K,T}^{(j)}, w_{K,T}^{(j)})\}_{j=1,\ldots,m}$.

from the target distribution $\pi(\boldsymbol{x}_T; \kappa_K)$ are highly concentrated around the true optimal path $\boldsymbol{x}_T^*$ and hence are useful in inferring the most likely path.

In summary, annealed SMC provides an iterative procedure to the difficult sampling problem under κ_K by utilizing the samples obtained at a higher temperature. On the one hand, annealed SMC provides a relatively "flat" and easy-sampling starting distribution $\pi(\boldsymbol{x}_T; \kappa_0)$ and designs a slow-changing path connecting $\pi(\boldsymbol{x}_T; \kappa_0)$ to the desired "sharp" distribution $\pi(\boldsymbol{x}_T; \kappa_K)$. On the other hand, for each iteration $k = 1, \ldots, K$, annealed SMC adopts an optimal proposal distribution $p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_T; \kappa_{k-1})$, which incorporates the full information set $\boldsymbol{y}_T$ and is usually difficult to evaluate in conventional SMC implementations. In annealed SMC, the proposal distribution is estimated by sample paths from the previ-

ous iteration. The details in estimating the proposal distribution will be discussed in the supplementary material.

Our annealing framework falls into the general framework of simulated annealing. The design of temperature sequences $\{\kappa_k\}_{k=0,\ldots,K}$ is known as the "cooling schedule". Kirkpatrick et al. (1983) uses an exponential schedule such that $\kappa_k = \alpha^k \kappa_0$ for some positive number α. A more conservative schedule such that $\kappa_k \propto \log(1+k)$ is suggest by Hajek (1988) and Aarts and Korst (1989) to ensure convergence to global minimum. Ingber (1989) proposed a fast adaptive cooling schedule that allows the temperature to increase (or κ to decrease) in order to re-gain broadness of samples at certain point. The specific choice of cooling schedule is beyond the scope of this manuscript. By default, we will choose the most aggressive exponential schedule with a picked value of α for faster convergence in the example section and the results are promising.

The conventional simulated annealing algorithm (Kirkpatrick et al., 1983) is a variation of Markov Chain Monte Carlo (MCMC), which adapts Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) with an extra temperature control. The convergence of the conventional simulated annealing algorithm is given by Granville et al. (1994). However, different from the conventional simulated annealing, annealed SMC does not require for a mixing condition as usually shown in MCMC algorithms. At each iteration at κ_k, the samples are always properly weighted with respect to the target distribution $\pi(\boldsymbol{x}_T; \kappa_k)$ because of the weight adjustments. The convergence of SMC is discussed in Crisan and Doucet (2000).

The terminology "annealed SMC" was also used in the literature of Ulker et al. (2011) and Wang et al. (2019), which are different from our method. The method of Ulker et al. (2011) and Wang et al. (2019) (we hereby refer it as "SMC annealing") constructs an annealing sequence of intermediate target distributions $\pi_t(\boldsymbol{x})$ indexed by $t = 0, \ldots, T$ with $\pi_0(\boldsymbol{x})$ as the beginning distribution and $\pi_T(\boldsymbol{x})$ as the terminating distribution, with the goal of generating a set of samples following the terminating distribution, by starting from samples following a relatively flat beginning distribution. SMC techniques are used when translating samples from the current distribution $\pi_t(\boldsymbol{x})$ to the next $\pi_{t+1}(\boldsymbol{x})$ by adopting a MCMC move as the proposal distribution. Our method also constructs a sequential of annealed target distributions $\pi_k(\boldsymbol{x}_T \mid \kappa_k)$, with the goal of optimization via Monte Carlo of a (near) degenerated terminating distribution. In our method, within each temperature (κ_k), SMC is utilized to sample the high dimensional $\boldsymbol{x}_T$ under a dynamic system set-up. The sequence of SMC proposal distributions within each temperature use the information contained in the Monte Carlo samples from the previous temperature.

More specifically, there are three major differences between ours and the SMC annealing method. First, the goal of SMC annealing is to draw samples from a target distribution (usually the posterior) that is difficult to directly sample from. The goal of our algorithm is to find the optimum such that the terminating distribution is proportional to the original one raised to an arbitrarily high power. Second, our method solves the problem when $\boldsymbol{x}$ itself is high-dimensional with a dynamic structure for which SMC is used to sequentially sample the components of $\boldsymbol{x}$, while SMC annealing deals with relatively lower dimensional $\boldsymbol{x}$ without

the need of SMC sampling. Third, SMC annealing uses SMC on the sequence of annealing distributions, while our method performs SMC within each annealing temperature and uses samples from the last iteration for constructing the internal SMC propagation proposal step in the subsequent temperature.

3.4 Path refinement with Viterbi algorithm

A more accurate estimate of the mode can be obtained by using Viterbi algorithm (Viterbi, 1967) on the discrete space consisting of the SMC samples. Viterbi algorithm is a dynamic programming algorithm originally used to solve the MLP problem in hidden Markov models, where the hidden states are finite. Let $\mathcal{A}_t = \{a_t^{(j)}\}_{j=1,\ldots,m}$ be the grid points for x_t and $\Omega = \mathcal{A}_1 \times \cdots \times \mathcal{A}_T$ be the Cartesian product of the grid point sets. In state space models, the Viterbi algorithm searches for the maximum over all possible combinations of the grid points in Ω. Specifically, the MLP obtained by the Viterbi algorithm is

$$\hat{\boldsymbol{x}_T}^{(viterbi)} = \arg\max_{\boldsymbol{x}_T \in \Omega} \pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T). \tag{3.17}$$

The Viterbi algorithm for state space models based on the grid points $\{a_1^{(j)}\}_{j=1,\ldots,m}, \ldots,$ $\{a_T^{(j)}\}_{j=1,\ldots,m}$ is depicted in Figure 3.

Although, the original Viterbi algorithm is designed for discrete state spaces, we adopt it to continuous state spaces by discretizing the state space to a set of selected finite grid points at each time point. The performance depends on the "quality" of the selected grid points (for example, how densely close to the underlying optimal path) and the number of grid points used. Here we use the generated Monte Carlo samples as the discretizing grid points.

Figure 3: Viterbi Algorithm for Markovian State Space Models

- *Let* $\mathcal{A}_t = \{a_t^{(j)}\}_{j=1,\ldots,m}$ *be a set of grid points for* x_t *for* $t = 1, \ldots, T$.
- *At time 1, initialize* $\ell_0^{(j)} = 0$ *and* $\hat{\boldsymbol{x}}_1^{(j)} = a_1^{(j)}$ *for* $j = 1, \ldots, m$.
- *At each time* $t = 2, \ldots, T$, *for* $j = 1, \ldots, m$, *set*

$$\ell_t^{(j)} = \max_{k \in \{1,\ldots,m\}} \ell_{t-1}^{(k)} p_t(a_t^{(j)} \mid \hat{\boldsymbol{x}}_{t-1}^{(k)}) g_t(y_t \mid a_t^{(j)}), \tag{3.18}$$

  and set $\hat{\boldsymbol{x}}_t^{(i)} = (\hat{\boldsymbol{x}}_{t-1}^{(k_j^*)}, a_t^{(j)})$, *where* j_j^* *is the optimal point of (3.18).*
- *Return* $\hat{\boldsymbol{x}}_T^{(j^*)}$, *where* $j^* = \arg\max_{j \in \{1,\ldots,m\}} \ell_T^{(j)}$.

As these samples follow the target distribution at low temperature, they should concentrate in the important regions.

For example, one can set $\mathcal{A}_t = \{x_t^{(i)}\}_{i=1,\ldots,m}$ such that $\Omega = \{x_1^{(i)}\}_{i=1,\ldots,m} \times \cdots \times \{x_T^{(i)}\}_{i=1,\ldots,m}$ is the joint set of all SMC sample points. Running Viterbi algorithm through these samples can only improve the result from the Monte Carlo samples, but will not obtain the underlying optimal path in the continuous space. Therefore, we call this step "refinement" instead of "optimization".

One can also add and remove grids points to expand coverage with more details around the more important state paths. For instance, in the lasso regression example in the supplement material, a Viterbi refinement helps to shrink the estimate of the zero-coefficients to

exactly 0.

The Viterbi algorithm explores all combination of sample points and results in a better mode estimation compared with the empirical MAP in (3.13). However, it has its limitations for implementation with state space models. One limitation is that the Viterbi algorithm only works on Markovian state space models. In addition, it only works with a non-singular state evolution in which the degree of freedom is the same as the state variable dimension. Otherwise, state paths cannot be re-assembled as Viterbi algorithm tries to achieve. For example, in the cubic spline problem, the state evolution is singular. Although one can reduce the dimension of the state variable to make the evolution non-singular, the state evolution then becomes non-Markovian. Another limitation is the requirement for Monte Carlo sample size. The Monte Carlo samples induced Ω provide a discretization of the support $\mathcal{X}$ for each time t. The accuracy of the Viterbi algorithm strongly dependents on the discretization quality, especially when $\mathcal{X}$ is continuous. In general, the denser the Monte Carlo samples are around the true MLP, the more accurate the Viterbi algorithm solution is. As a result, it often requires a large Monte Carlo sample size to generate better discretization and to achieve high accuracy. To reduce the path error $\|\hat{x}_{1:T}^{(viterbi)} - x_{1:T}^*\|$ by half, the Monte Carlo sample size m needs to be doubled, because the discretization size is reduced by half on average with doubled sample size. On the other hand, the computational cost increases quadratically with the sample size m. One possible way to improve is to apply Viterbi algorithm iteratively by shrinking to the high value region of last iteration and regenerating grid points there. Similar to iterative grid search, the iterative Viterbi algorithm may get a

sub-optimal solution.

4. Simulation Results

In this section, we provide the simulated results of annealed SMC in finding the most likely path for the two emulated state space modes from Section 2.3.

Note that, the smoothing spline problem has a close-form solution. Even in the emulated state space model setting, Kalman filter can give the exact solution. It is used for illustration purpose only. On the other hand, the optimal trading path problem is not trivial and is a real application that the proposed method is ideally suitable, especially when non-linear solvers usually give less accurate solutions.

Two more examples are demonstrated in the supplementary material. We aim to demonstrate the flexibility of the proposed method in solving existing optimization problems with some new applications, though our approach may not yield better performance than specially designed optimization algorithms for general problems.

4.1 Cubic Smoothing Spline

In this simulation study, we consider the cubic smoothing spline problem in Section 2.3.1. The observations are generated by $y_t = \sin(9(t-1)/100) + \zeta_t$, for $t = 1, \ldots, 50$, with $\zeta_t \sim \mathcal{N}(0, 1/16)$ and we fix $\lambda = 10$ in the objective function (2.7).

Since the dynamic system is linear and Gaussian, the most likely path is obtained by Kalman Smoother (Kalman, 1960). We use it as the benchmark. We start from the initial

inverse temperature $\kappa = \kappa_0 = 4$. Figure 4 demonstrates $m = 1000$ samples (in grey) drawn from the target distribution $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T; \kappa_0) \propto [\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T)]^{\kappa_0}$ by the SMC algorithm described in Figure 1 along with the observations $\boldsymbol{y}_T$ (the solid line) and the true most likely path (the dashed line).

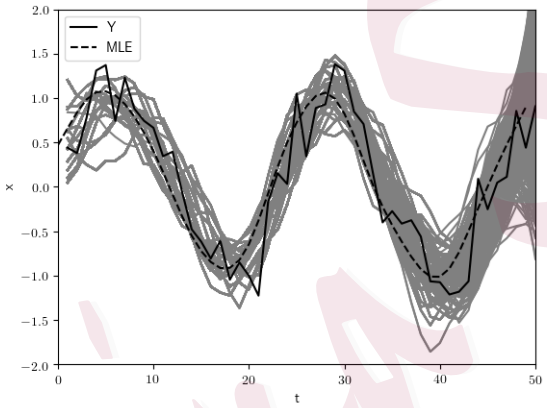

Figure 4: Sample paths at $\kappa_0 = 4$.

The proposal distribution $q_t(\cdot)$ used at κ_0 is chosen to be proportional to $p_t(x_t \mid \boldsymbol{x}_{t-1}) g_t(y_t \mid x_t)$. At each time t, η_t is drawn from the proposal distribution $q_t(\eta_t \mid a_{t-1}, b_{t-1}, c_{t-1}, y_t)$, which is Gaussian. Resampling is conducted when the effective sample size (ESS) defined in (4.19) is less than $0.3m$.

$$ESS = \frac{(\sum_{i=1}^{m} w_t^{(i)})^2}{\sum_{i=1}^{m} (w_t^{(i)})^2}. \tag{4.19}$$

To find the most likely path stochastically and numerically, we apply the annealed SMC approach in Figure 2 with a predetermined sequence of inverted temperatures $\kappa_k = 1.5^k \kappa_0$ for $k = 1, \ldots, 16$. The proposal distribution for the anneal SMC is estimated by the parametric

approach (see the supplementary material). Specifically, since the innovation in the state equation is of one dimension, at κ_k, we only need to generate proposal samples for c_t. It is drawn by first fitting $\{(c_{k-1,t}^{(j)}, a_{k-1,t-1}^{(j)}, b_{k-1,t-1}^{(j)}, c_{k-1,t-1}^{(j)})\}_{j=1,\ldots,m}$ with a multivariate Gaussian distribution and then sampling from the conditional distribution. To prevent degeneracy, resampling step is only conducted at the end of each annealed SMC iteration and after each iteration, one step of post-MCMC move is conducted to regenerate sample states. The post-MCMC move uses blocked Gibbs sampling (Jensen et al., 1995), due to the special structure of the state dynamic. At each iteration of the Gibbs sampling, (x_t, x_{t+1}, x_{t+2}) are updated together.

Figure 5 shows the sample paths (after the post-MCMC step) at the end of different annealed SMC iterations. When the temperature shrinks to zero as κ increases, the sample paths move to a small neighborhood region around the true most likely path. Figure 6 shows the value of the objective function at the weighted average path of the samples as for different numbers of iterations. The true optimal value (the objective function value at the optimal path) obtained by the Kalman smoother is plotted as the dashed horizontal line. As the number of iteration increases, the objective function value at the averaged path decreases stochastically and convergences at roughly the 7th iteration.

To compare the computational efficiency, we record the computing time needed for different approaches as follows. Kalman smoother takes 2.2ms, Scipy minimizer takes 129.6ms and annealed SMC takes 232.9ms. The Scipy approach uses the nonlinear optimizer provided by the python package Scipy (Jones et al., 2001), which implements the Broyden-Fletcher-

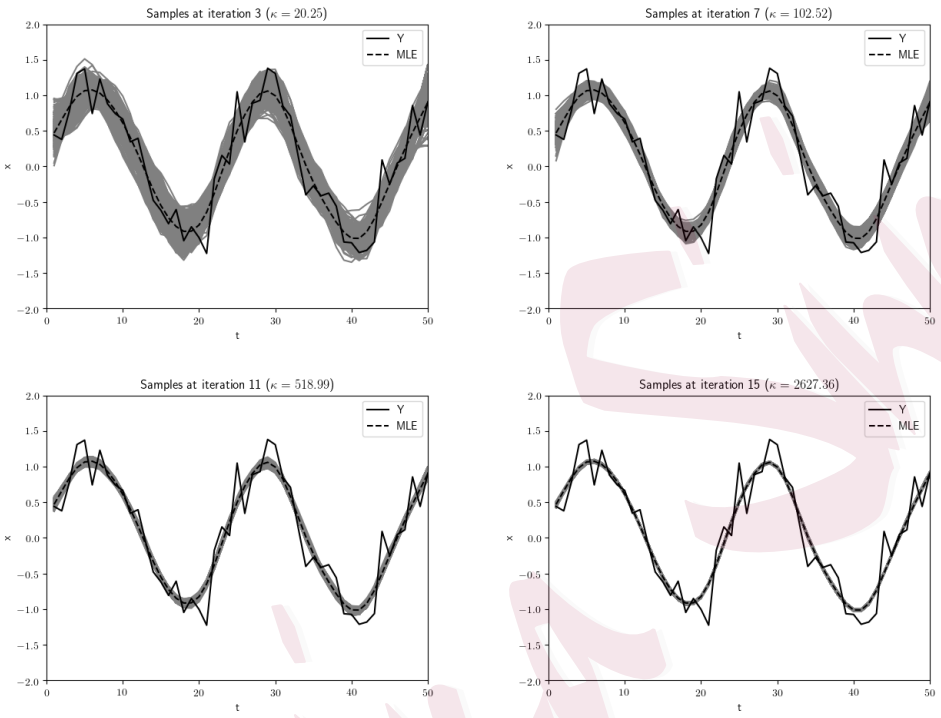

Figure 5: Sample paths at different κ's

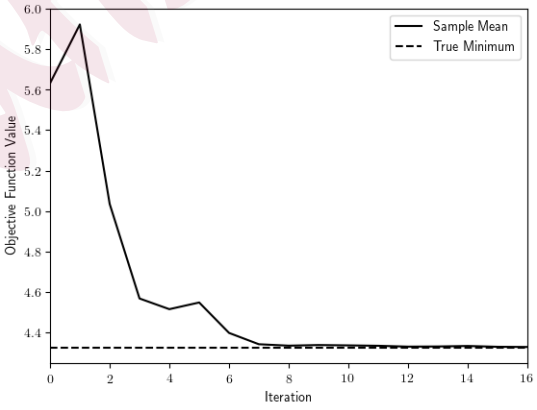

Figure 6: Value of the objective function against the number of iterations

Goldfarb-Shanno (BFGS) algorithm by default. The annealed SMC records the time until convergence (the time when the value of objective function is not improved by further iteration). Kalman Smoother is the fastest one due to its deterministic nature in finding the most likely path for linear Gaussian models. Annealed SMC is slower than the nonlinear solver program provided by Scipy, but achieves similar accuracy. We also note that this is a simple convex optimization problem in which a straightforward optimization algorithm such as the Scipy performs well. Our estimation approach is more flexible and this example serves as an illustration of how the algorithm works.

4.2 Optimal Trading Path

In this simulation, we consider the optimal trading path problem in Section 2.3.2. Following Cai et al. (2018), we set $T = 20$, $\sigma_x^2 = 0.25$, $\sigma_y^2 = 1$ and $\alpha = 0.5$. The ideal trading path is given by

$$y_t = 25\exp\{-(t+1)/8\} - 40\exp\{-(t+1)/4\}.$$

We start from the initial temperature $\kappa = \kappa_0 = 1.0$. The sample paths at κ_0 are drawn with the constrained SMC (Cai et al., 2018), where the resampling step is conducted with priority scores $\beta_t(\boldsymbol{x}_t) \propto \hat{p}(y_{t+1}, \ldots, y_T \mid x_t)$. The priority scores are estimated from a set of backward pilot samples (Cai et al., 2018). In this example, we use $m^* = 300$ backward pilot samples. The resulting $m = 1000$ (forward) sample paths are shown in Figure 7. The observations $y_1, \ldots, y_T$, which represent the ideal optimal trading strategy without the trading cost, are plotted as the solid line. An estimated path (dashed line) is provided by

the Scipy nonlinear optimization algorithm.

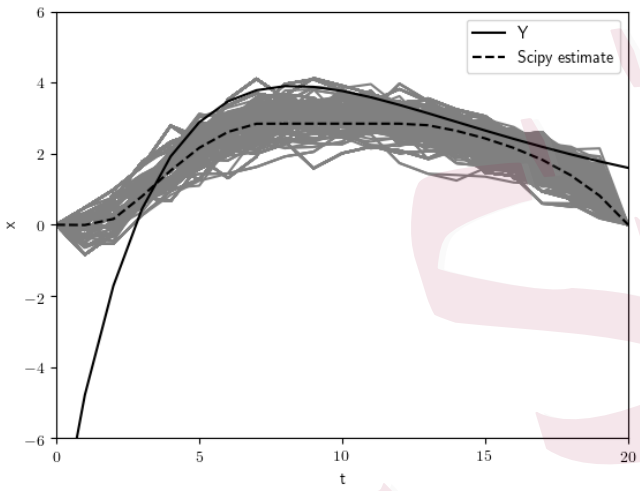

Figure 7: Sample paths at κ_0

We use the following sequence of inverted temperatures for annealing: $\kappa_k = 2^k \kappa_0$ for $k = 1, \ldots, 20$. The proposal distribution in the annealed SMC is sampled with the parametric approach by approximating the joint distribution of $x_{k-1,t}$ and $x_{k-1,t-1}$ with a bivariate normal distribution. The annealed $m = 1000$ sample paths are resampled at the end of each iteration, and no post-MCMC step is conducted. Samples at several different inverted temperatures are shown in Figure 8. We use the sample average as our estimator for the most likely path. The value of the objective function at the sample average path decreases stochastically as shown in Figure 9. It eventually converges at around the 11th iteration. The optimal objective function value achieved by the annealed SMC is 89.459, while the one obtained by the Scipy nonlinear optimizer is 89.462. The values of the objective function at

the sample paths at the 20th iteration has an average of 89.459 and a standard deviation of 1.09×10^{-5}. The annealed SMC gains some improvement in accuracy at the cost of extra computation. The Scipy nonlinear optimizer takes 78ms while the annealed SMC costs 1.820 seconds for the initial emulated model (including the time of backward sampling) and costs around 2ms for each subsequent iteration. Sampling from the base emulated model costs much more than subsequent iteration for two reasons. First, it requires a large sample size for the base model because of high degeneracy. Second, the end point constraint is imposed and an additional backward pilot run is needed to reduce degeneracy.

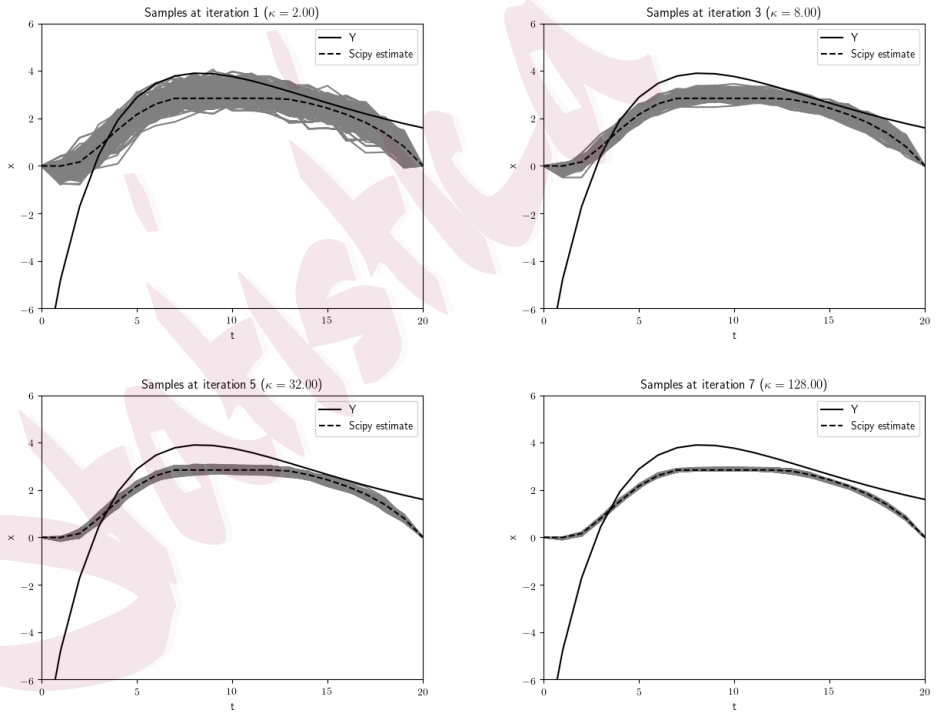

Figure 8: Sample paths at different κ's

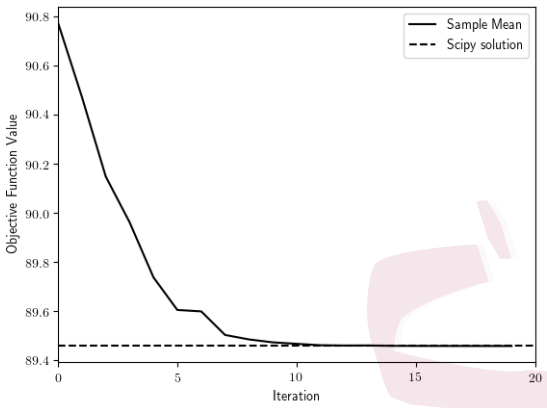

Figure 9: Value of the objective function against the number of iterations

5. Summary and Discussion

In this article, we propose a general framework of state space model emulation for high dimensional optimization problems. The main idea of emulation is to change the goal from optimization to sampling. We demonstrated that, by constructing a proper state space model, many high dimensional optimization problems can be turned into the problem of finding the optimal (most likely) path under the state space model. In order to reduce the accuracy lose due to the sampling nature, we proposed the annealing steps with an extremely sharp terminating distribution, where the samples, though random, are highly concentrated around the optimum (the most likely path) We demonstrate the procedure of state space model emulation with two conventional problems in the main content and additional two in the supplementary material and show how they can be solved using the proposed annealed SMC approach.

The proposed annealed SMC approach shares some similar properties with traditional simulated annealing methods. Both can optimize a wide range of objective functions including non-convex functions and multi-modal functions, and both often require heavier computation cost than the simpler standard optimization algorithms such as the gradient descent algorithms. However, the annealed SMC approach for state space models is different to the traditional simulated annealing methods in association with MCMC for stochastic optimization in the following ways. First, emulating an optimization problem into a state space model has its advantage in many problems, especially when the problem is of high dimensional and when the system is inherently dynamic (such as the trading path problem or the ℓ_1 trend filtering problem) or when the parameters to be estimated inherently play similar roles in the problem (such as the parameters in the regression problem). Second, SMC as an alternative to MCMC has certain advantages in many fixed dimensional problems such as in the problems when the "dependence" between the parameters in the emulated target distribution is local and (locally) very strong. In these problems, MCMC encounters slow mixing difficulties while SMC naturally takes advantage of such properties. Third, given any temperature, SMC samples target the equilibrium distribution, while MCMC samples often move towards the target distribution gradually. Hence annealed SMC may tolerate faster cooling schedule. Fourth, the inherited parallel structure of SMC allows faster computation. It also adapts to multi-modal problems better.

The state space model emulation and the annealed SMC provide an alternative way to solve high-dimensional optimization problems. Of course, the approach may not be suitable

for all problems, due to its high computational cost and its requirement of certain structures. Nevertheless the approach adds to the high dimensional optimization toolbox a useful method for a wide range of complex problems for which the more traditional method may have difficulties to solve. Although the examples shown in this paper do not demonstrate great improvement of the state space emulation approach over the traditional one, they have effectively shown how the approach can be implemented and can be used for other problems.

Supplementary Materials

Discussion on technical details in the annealed SMC algorithm and two additional emulation examples with simulation results are provided in the supplementary material.

Acknowledgements

The authors thank the editor, associate editor, and two anonymous referees for their helpful comments and suggestions. Chen's research is supported in part by National Science Foundation grants DMS-1737857, DMS-2052949, DMS-2027855, CCF-1934924 and IIS-1741390.

References

Aarts, E. and J. Korst (1989). *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing*. John Wiley & Sons, Inc.

Agarwal, A., A. Anandkumar, P. Jain, P. Netrapalli, and R. Tandon (2014). Learning sparsely used overcomplete dictionaries. In *Conference on Learning Theory*, pp. 123–137.

Anandkumar, A., R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky (2014). Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research 15*(1), 2773–2832.

Arora, S., R. Ge, R. Kannan, and A. Moitra (2012). Computing a nonnegative matrix factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 145–162. ACM.

Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society 48*(3), 334–334.

Briers, M., A. Doucet, and S. Maskell (2010). Smoothing algorithms for state-space models. *Annals of the Institute Statistical Mathematics 62*, 61–89.

Cai, A., R. S. Tsay, and R. Chen (2009). Variable selection in linear regression with many predictors. *Journal of Computational and Graphical Statistics 18*(3), 573–591.

Cai, C., R. Chen, and M. Lin (2018). Resampling strategy in sequential monte carlo for constrained sampling problems. *arXiv preprint arXiv:1706.02348*.

Chen, R., X. Wang, and J. S. Liu (2000). Adaptive joint detection and decoding in flat-fading channels via mixture kalman filtering. *IEEE Transactions on Information Theory 46*(6), 2079–2094.

Crisan, D. and A. Doucet (2000). Convergence of sequential monte carlo methods. *Signal Processing Group, Department of Engineering, University of Cambridge, Technical Report CUEDIF-INFENGrrR38 1*.

Del Moral, P., A. Doucet, and S. Singh (2010). Forward smoothing using sequential monte carlo. *arXiv preprint arXiv:1012.5390*.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1–38.

Doucet, A., N. De Freitas, and N. Gordon (2001). An introduction to sequential monte carlo methods. In *Sequential*

Monte Carlo Methods in Practice, pp. 3–14. Springer.

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.

Godsill, S. J., A. Doucet, and M. West (2004). Monte carlo smoothing for nonlinear time series. *Journal of the American Statistical Association 99*(465), 156–168.

Gordon, N. J., D. J. Salmond, and A. F. Smith (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing) 140*(2), 107–113.

Granville, V., M. Krivanek, and J. . Rasson (1994, June). Simulated annealing: a proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence 16*(6), 652–656.

Green, P. J. and B. W. Silverman (1993). *Nonparametric regression and generalized linear models: a roughness penalty approach*. CRC Press.

Guarniero, P., A. M. Johansen, and A. Lee (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association 112*(520), 1636–1647.

Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of operations research 13*(2), 311–329.

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

Hooke, R. and T. A. Jeeves (1961, April). " direct search" solution of numerical and statistical problems. *J. ACM 8*(2), 212–229.

Iglesias, M. A., K. J. H. Law, and A. M. Stuart (2013, mar). Ensemble kalman methods for inverse problems. *Inverse Problems 29*(4), 045001.

Ingber, L. (1989). Very fast simulated re-annealing. *Mathematical and Computer Modelling 12*(8), 967–973.

Irie, K. and M. West (2016). Bayesian emulation for optimization in multi-step portfolio decisions. *arXiv preprint arXiv:1607.01631*.

Jain, P., P. Kar, et al. (2017). Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning 10*(3-4), 142–336.

Jensen, C. S., U. Kjærulff, and A. Kong (1995). Blocking gibbs sampling in very large probabilistic expert systems. *International Journal of Human-Computer Studies 42*(6), 647–666.

Jones, E., T. Oliphant, P. Peterson, et al. (2001). SciPy: Open source scientific tools for Python.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering 82*(1), 35–45.

Kiefer, J., J. Wolfowitz, et al. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics 23*(3), 462–466.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi (1983). Optimization by simulated annealing. *Science 220*(4598), 671–680.

Kitagawa, G. (1996). Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics 5*(1), 1–25.

Kolm, P. N. and G. Ritter (2015). Multiperiod portfolio selection and bayesian dynamic models. *Risk*. Available at SSRN: https://ssrn.com/abstract=2472768.

Kong, A., J. S. Liu, and W. H. Wong (1994). Sequential imputations and bayesian missing data problems. *Journal of the American statistical association 89*(425), 278–288.

Kyle, A. and A. Obizhaeva (2011). Market microstructure invariants: Theory and implications of calibration. Available at SSRN: https://ssrn.com/abstract=1978932.

Land, A. H. and A. G. Doig (1960). An automatic method of solving discrete programming problems. *Econometrica 28*(3), 497–520.

Lin, M., R. Chen, and J. S. Liu (2013). Lookahead strategies for sequential Monte Carlo. *Statistical Science 28*, 69–94.

Lin, M. T., J. L. Zhang, Q. Cheng, and R. Chen (2005). Independent particle filters. *Journal of the American Statistical Association 100*(472), 1412–1421.

Liu, J. S. and R. Chen (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association 90*(430), 567–576.

Liu, J. S. and R. Chen (1998). Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association 93*(443), 1032–1044.

Markowitz, H. (1959). Portfolio selection, cowles foundation monograph no. 16. *John Wiley, New York 32*, 263–74.

Mei, S., A. Montanari, and P.-M. Nguyen (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences 115*(33), E7665–E7671.

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.

Nelder, J. A. and R. Mead (1965). A simplex method for function minimization. *The computer journal 7*(4), 308–313.

Netrapalli, P., U. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain (2014). Non-convex robust pca. In *Advances in Neural Information Processing Systems*, pp. 1107–1115.

Pitt, M. K. and N. Shephard (1999). Filtering via simulation: auxiliary particle filters. *Journal of the American statistical association 94*(446), 590–599.

Ulker, Y., B. Gunsel, and A. T. Cemgil (2011). Annealed smc samplers for nonparametric bayesian mixture models. *IEEE Signal Processing Letters 18*(1), 3–6.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory 13*(2), 260–269.

Wang, L., S. Wang, and A. Bouchard-Côté (2019, 06). An Annealed Sequential Monte Carlo Method for Bayesian Phylogenetics. *Systematic Biology 69*(1), 155–183.

Zhang, P., Q. Song, and F. Liang (2021). A langevinized ensemble kalman filter for large-scale static and dynamic learning. *arXiv preprint arXiv:2105.05363*.

Chencheng Cai

Department of Statistics, Operations and Data Science, Fox School of Business,

Temple University, Philadelphia, PA 19122, USA

E-mail: chencheng.cai@temple.edu

Rong Chen

Department of Statistics, Rutgers University, Piscataway, NJ 08854, USA

E-mail: rongchen@stat.rutgers.edu