状態の後退遷移方程式を用いたモンテカルロ粒子平滑化とフィルター初期化*

矢野浩一†

概　要

本論文では状態の後退遷移方程式（後退方程式）を用いたモンテカルロ粒子フィルターに基づく平滑化アルゴリズムとフィルタリングの初期化アルゴリズムについて提案する。本論文の方式は後退方程式が解析的に得られる場合において非線形非ガウス状態空間モデルにおける状態平滑化とフィルター初期化を実現するものである。本論文で提案する平滑化はモンテカルロ粒子フィルターとほぼ同等のアルゴリズムで実現できるため、その計算量はモンテカルロ粒子フィルターの計算量と同等である。さらに、本論文では非線形非ガウス状態空間モデルによるフィルタリングの初期化アルゴリズムを提案する。このアルゴリズムは本論文での平滑化アルゴリズムと後退方程式を用いて実現される。本論文では提案手法の有効性を示すため、線形ガウス状態空間モデル、線形非ガウス状態空間モデル、確率的ボラティリティ変動モデル、t-分布付確率的ボラティリティ変動モデルによるシミュレーションを実施する。

キーワード： 時系列解析, モンテカルロ粒子フィルター, 非線形非ガウス状態空間モデル, 平滑化アルゴリズム, 確率的ボラティリティ変動モデル.

---

*本論文に有益なご意見を下さった北川源四郎教授（統計数理研究所所長 [大学共同利用機関法人 情報・システム研究機構]）、田村義保教授（統計数理研究所副所長）、佐藤整尚准教授（統計数理研究所）、中村和幸博士（統計数理研究所）に感謝します。本論文では統計数理研究所の ALTIX3700 スーパーコンピュータシステムをシミュレーションに使用した。なお本稿の内容は全て著者の個人的見解であり、金融庁及び金融研究研修センターの公式見解ではない。

†金融庁金融研究研修センター研究官　E-mail: koichi.yano@fsa.go.jp.

The Monte Carlo Particle Smoothing and Filter Initialization Based on The Backward Transition of States *

Koiti Yano†

Abstract

This paper proposes a smoothing algorithm based on the Monte Carlo Particle filter and the backward transition equation of states (a backward equation). Our method is applicable to any nonlinear non-Gaussian state space model if a backward equation is given analytically. The computational complexity of our smoothing algorithm is equal to the complexity of the Monte Carlo Particle filter because it can be realized by a minor modification of the Monte Carlo Particle filter. Moreover, we propose a filter initialization algorithm based on the smoothing distribution which is obtained by our smoothing algorithm and a backward equation. In this paper, we demonstrate the effectiveness of our method by applying it to a linear Gaussian state space model, a linear non-Gaussian state space model, a stochastic volatility model, and a stochastic volatility model with a t-distribution.

Key words : Time Series Analysis, Monte Carlo Particle Filter, Nonlinear non-Gaussian State Space Model, Smoothing Algorithm, Stochastic Volatility Model.

---

*The author would like to thank to Genshiro Kitagawa, Kazuyuki Nakamura, Seisho Sato, and Yoshiyasu Tamura for their helpful comments. The author would like to thank the Institute of Statistical Mathematics for the facilities and the use of SGI Altix3700. This paper presents the author's personal views, which are not necessarily the official ones of the Financial Research and Training Center or the Financial Services Agency.

†Research Fellow, Financial Services Agency, Government of Japan. E-mail: koichi.yano@fsa.go.jp

Contents

1 Introduction

The Monte Carlo Particle (MCP) filter was proposed by Gordon et al. (1993) and Kitagawa (1996). The filter is an algorithm to estimate states for a nonlinear non-Gaussian state space model [3]. In recent years, the filter has been applied to various problems [4]. In spite of the widespread practical application of the MCP filter, smoothing algorithms are less well established. The first smoothing algorithm, proposed by Kitagawa (1996), is based on the storing state vector. In that algorithm, the repetition of the resampling in the MCP filter decreases the number of different realizations of the state vector. To resolve that problem, Kitagawa (1996) proposes the employment of fixed L-lag smoothing. The paper recommends not making L too large (say, 10 or 20, or 50 at the largest). A persistent problem is that the fixed L-lag smoothing cannot use all observations when the number of observations is larger than 10–50. To realize fixed-interval smoothing, researchers have developed alternative methods based on a recursive recomputation approach (Nakamura and Tsuchiya (2007)), the two-filter formula (Kitagawa (1996)), a new generalized two-filter formula (Briers et al. (2004)), a forward filtering – backward smoothing formula (Doucet et al. (2000) and Godsill et al. (2004)), and maximum $a\ posteriori$ sequence estimation (Godsill et al. (2001).)

This paper proposes a simple MCP smoothing algorithm based on the backward transition equation of states (a backward equation). Our method is applicable to any nonlinear non-Gaussian state space model if a backward equation is given analytically. This paper shows that our algorithm is fundamentally equivalent to the MCP filter. Therefore, the computational complexity of our smoothing algorithm is equal to the complexity of the MCP filter.

[3] Arulampalam et al. (2002) is a readable tutorial on the Monte Carlo particle filter.

[4] see Doucet et al., eds (2001).

Moreover, our algorithm is easily implemented because it can be realized by a minor modification of the MCP filter. The main advantage of our algorithm is its simplicity. Furthermore, we propose a filter initialization algorithm based on the smoothing distribution, which is obtained by our algorithm and a backward equation. Filter initialization is important to estimate the state vectors of nonlinear non-Gaussian state space models in Bayesian tracking like the MCP filter. In this paper, we show the effectiveness of our method by applying it to a linear Gaussian state space model, a linear non-Gaussian state space model, a stochastic volatility model, and a stochastic volatility model with a t-distribution.

Our algorithm is inspired by Klaas et al. (2006), which proposes a fast particle smoothing algorithm. Their algorithm is a smoothing algorithm based on a new generalized two-filter smoother, proposed by Briers et al. (2004). However, our algorithm is based on Eq. (11), which is described in section 2.2. We would like to emphasize that our algorithm is simpler than the fast particle smoothing algorithm and it requires less memory rather than the their method.

This paper is organized as follows. In section 2, we describe our smoothing algorithm and filter initialization algorithm. In section 3, we show examples for some models. In section 4, we describe the salient conclusions of the paper.

2 Model

2.1 Monte Carlo Particle Filter

A nonlinear non-Gaussian state space model for the time series $\boldsymbol{y}_t,\ t = \{1, 2, \cdots, T\}$ is defined as

$$\begin{aligned} \boldsymbol{x}_t &= f(\boldsymbol{x}_{t-1}, \boldsymbol{v}_t), \\ \boldsymbol{y}_t &= h(\boldsymbol{x}_t, \boldsymbol{\epsilon}_t), \end{aligned} \tag{1}$$

where $\boldsymbol{x}_t$ is an unknown $n_x \times 1$ state vector, $\boldsymbol{v}_t$ is the $n_v \times 1$ system noise vector with a density function $q(\boldsymbol{v})$, $\boldsymbol{\epsilon}_t$ is the $n_\epsilon \times 1$ observation noise vector with a density function $r(\boldsymbol{\epsilon})$. The function $f : \boldsymbol{R}^{n_x} \times \boldsymbol{R}^{n_v} \to \boldsymbol{R}^{n_x}$ and the function $h : \boldsymbol{R}^{n_x} \times \boldsymbol{R}^{n_\epsilon} \to \boldsymbol{R}^{n_y}$ are possibly nonlinear functions. The first equation of (1) is called a system equation and the second equation is called a measurement equation. This nonlinear non-Gaussian state space model specifies the following two conditional density functions.

$$\begin{aligned} &p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \\ &p(\boldsymbol{y}_t|\boldsymbol{x}_t). \end{aligned} \tag{2}$$

The MCP filter is a variant of sequential Monte Carlo algorithms. In the MCP filter, a posterior density function is approximated as "particles" that have weights, as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{1}{\sum_{i=1}^{M} w_t^i} \sum_{i=1}^{M} w_t^i \delta(\boldsymbol{x}_t - \boldsymbol{x}_t^i), \tag{3}$$

where w_t^i is the weight of a particle $\boldsymbol{x}_t^i$, M is the number of particles, $\boldsymbol{y}_{1:t}$ is $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_t\}$, and $\delta(\cdot)$ is the Dirac delta function [5]. Particles $\boldsymbol{x}_t^i$ are sampled from the system equation of Eq. (2). The weight w_t^i is defined as

$$w_t^i = r(\psi(\boldsymbol{y}_t, \boldsymbol{x}_t^i))\left|\frac{\partial \psi}{\partial \boldsymbol{y}_t}\right|, \tag{4}$$

where ψ is the inverse equation of the function h [6]. The right hand side (RHS) of Eq. (4) represents the likelihood function of a nonlinear non-Gaussian state space model. In the standard algorithm of the MCP filter, the particles x_t^i are resampled with sampling probabilities proportional to $w_t^1, \cdots, w_t^M$. Resampling algorithms are discussed in Kitagawa (1996). After resampling, we have $w_t^i = 1/M$. Consequently, Eq. (3) is rewritten as

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{1}{M} \sum_{i=1}^{M} \delta(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t^i),\ i = \{1, \cdots, M\}, \tag{5}$$

where $\hat{\boldsymbol{x}}_t^i$ are particles after resampling. The algorithm of the MCP filter is shown as Algorithm 1.

Algorithm 1: The Monte Carlo Particle Filter

$[\{\hat{\boldsymbol{x}}_t^i,\ w_t^i\}_{i=1}^M$, llk] = MCPfilter$[\{\hat{\boldsymbol{x}}_{t-1}^i\}_{i=1}^M,\ \boldsymbol{y}_t]$

$\{$

FOR i=1,...M

Predict: $\boldsymbol{x}_t^i \sim p(\boldsymbol{x}_t|\hat{\boldsymbol{x}}_{t-1}^i, v_t^i)$

Weight: $w_t^i = r(\psi(\boldsymbol{y}_t, \boldsymbol{x}_t^i))\left|\frac{\partial \psi}{\partial y}\right|$

ENDFOR

Sum of Weights: sw = $\sum_{i=1}^M w_t^i$

Log Likelihood: $llk = log(sw/M)$

FOR i=1,...,M

Normalize: $w_t^i = \frac{w_t^i}{sw}$

ENDFOR

[5] The Dirac delta function is defined as follow.

$$\delta(x) = 0,\ \text{if } x \neq 0,$$
$$\int_{\infty}^{\infty} \delta(x)dx = 1.$$

[6] See cite.

Resampling: $[\{\hat{\boldsymbol{x}}_t^i\}_{i=1}^M]$ =resample$[\{\boldsymbol{x}_t^i,\ w_t^i\}_{i=1}^M]$
RETURN$[\{\hat{\boldsymbol{x}}_t^i\}_{i=1}^M$, llk]
}
MCPmain $[\{\boldsymbol{x}_0^i\}_{i=1}^M,\ \{\boldsymbol{y}_t\}_{t=1}^T]$
{
Initialize: $llk = 0$
FOR t=1,...,T
mcp = MCPfilter$[\{\hat{\boldsymbol{x}}_{t-1}^i\}_{i=1}^M,\ \boldsymbol{y}_t]$
$llk = llk + (llk\ in\ \text{mcp})$
$\{\hat{\boldsymbol{x}}_t^i\}_{i=1}^M = (\{\hat{\boldsymbol{x}}_t^i\}_{i=1}^M\ in\ \text{mcp})$
ENDFOR
RETURN$[\{\{\hat{\boldsymbol{x}}_t^i\}_{i=1}^M\}_{t=1}^T, llk]$
}

2.2 Smoothing and Filter Initialization Based on a Backward Equation

We propose a smoothing algorithm based on a backward equation. Our algorithm is applicable to any nonlinear non-Gaussian state space model with a backward equation. We assume that a backward equation is given by

$$\boldsymbol{x}_{t-1} = g(\boldsymbol{x}_t, \boldsymbol{\xi}_s, \boldsymbol{v}_t), \tag{6}$$

where g is the backward transition equation of the state vector $\boldsymbol{x}_t$. Eq. (6) specifies the following conditional density function.

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t). \tag{7}$$

In sequential Monte Carlo methods, the smoothing distribution $p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T})$ is approximated by

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T}) = \frac{1}{\sum_{i=1}^M \tilde{w}_t^i} \sum_{i=1}^M \tilde{w}_t^i \delta(\boldsymbol{x}_{t+1} - \tilde{\boldsymbol{x}}_{t+1}^i). \tag{8}$$

The determination of $\tilde{w}_t^i$ is described below. After resampling like the MCP filter, the smoothing distribution $p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T})$ is given by

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T}) \simeq \frac{1}{M} \sum_{i=1}^M \delta(\boldsymbol{x}_{t+1} - \hat{\tilde{\boldsymbol{x}}}_{t+1}^i), \tag{9}$$

where $\hat{\tilde{x}}_{t+1}^i$ represent particles after resampling.

We derive the smoothing distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T})$ $(1 \leq t \leq T)$ from the definition of conditional probability.

If the probability distribution $\mathbb{P}(\boldsymbol{y}_{1:T}) > 0$, then the conditional probability of $\boldsymbol{x}_t$ given $\boldsymbol{y}_{1:T}$ is

$$\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}) = \frac{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}_{1:T})}{\mathbb{P}(\boldsymbol{y}_{1:T})}. \tag{10}$$

We define $\boldsymbol{y}_{1:T}^{(-t)} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{t-1}, \boldsymbol{y}_{t+1}, \cdots, \boldsymbol{y}_T\}$. We rewrite the conditional probability as follows.

$$\begin{aligned}
&\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:T})\\
=&\frac{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{y}_{1:T}^{(-t)})}{\mathbb{P}(\boldsymbol{y}_t, \boldsymbol{y}_{1:T}^{(-t)})}\\
=&\frac{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}_t, \boldsymbol{y}_{1:T}^{(-t)})}{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}_{1:T}^{(-t)})}\frac{\mathbb{P}(\boldsymbol{x}_t, \boldsymbol{y}_{1:T}^{(-t)})/\mathbb{P}(\boldsymbol{y}_{1:T}^{(-t)})}{\mathbb{P}(\boldsymbol{y}_t, \boldsymbol{y}_{1:T}^{(-t)})/\mathbb{P}(\boldsymbol{y}_{1:T}^{(-t)})}\\
=&\frac{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{y}_{1:T}^{(-t)})\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}^{(-t)})}{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{y}_{1:T}^{(-t)})}\\
=&\frac{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{x}_t)\mathbb{P}(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}^{(-t)})}{\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{y}_{1:T}^{(-t)})}.
\end{aligned} \tag{11}$$

In the fourth equality of Eq. (11), we assume that the likelihood, $\mathbb{P}(\boldsymbol{y}_t|\boldsymbol{x}_t)$, does not depend on $\boldsymbol{y}_{1:T}^{(-t)}$. The density, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}^{(-t)})$, is factorized as follows [7].

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}^{(-t)}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T}^{(-t)})d\boldsymbol{x}_{t+1}. \tag{13}$$

[7] This factorization is justified as follows.

$$\begin{aligned}
p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}^{(-t)}) &= \int p(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T}^{(-t)})d\boldsymbol{x}_{t+1}\\
&= \int \frac{p(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{y}_{1:T}^{(-t)})}{p(\boldsymbol{y}_{1:T}^{(-t)})}d\boldsymbol{x}_{t+1}\\
&= \int \frac{p(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}, \boldsymbol{y}_{1:T}^{(-t)})}{p(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{1:T}^{(-t)})}\frac{p(\boldsymbol{x}_{t+1}, \boldsymbol{y}_{1:T}^{(-t)})}{p(\boldsymbol{y}_{1:T}^{(-t)})}d\boldsymbol{x}_{t+1}\\
&= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}, \boldsymbol{y}_{1:T}^{(-t)})p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T}^{(-t)})d\boldsymbol{x}_{t+1}\\
&= \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T}^{(-t)})d\boldsymbol{x}_{t+1}
\end{aligned} \tag{12}$$

In the fifth equality of Eq. (12), we use the property that $\boldsymbol{x}_t$ depends only on $\boldsymbol{x}_{t+1}$ in Eq. (6).

If $p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T}^{(-t)})$ is nearly equal to $p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T})$, Eq. (13) can be rewritten

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}^{(-t)}) &\simeq \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:T})d\boldsymbol{x}_{t+1} \\ &\simeq \frac{1}{M}\sum_{i=1}^{M}\int p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})\delta(\boldsymbol{x}_{t+1}-\hat{\tilde{\boldsymbol{x}}}_{t+1}^i)d\boldsymbol{x}_{t+1} \\ &= \frac{1}{M}\sum_{i=1}^{M} p(\boldsymbol{x}_t|\hat{\tilde{\boldsymbol{x}}}_{t+1}^i) \\ &\simeq \frac{1}{M}\sum_{i=1}^{M}\delta(\boldsymbol{x}_t-\tilde{\boldsymbol{x}}_t^i), \end{aligned} \tag{14}$$

where particles $\boldsymbol{x}_t^i$ are sampled from Eq. (7). From Eq. (11) and (14), we obtain the following equation

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}) &\propto p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}^{(-t)}) \\ &\simeq \frac{1}{M}p(\boldsymbol{y}_t|\boldsymbol{x}_t)\sum_{i=1}^{M}\delta(\boldsymbol{x}_t-\tilde{\boldsymbol{x}}_t^i) \\ &= \frac{1}{M}\sum_{i=1}^{M}p(\boldsymbol{y}_t|\tilde{\boldsymbol{x}}_t^i)\delta(\boldsymbol{x}_t-\tilde{\boldsymbol{x}}_t^i). \end{aligned} \tag{15}$$

Eq. (15) indicates that the weight $\tilde{w}_t^i$ is obtained as

$$\tilde{w}_t^i \propto p(\boldsymbol{y}_t|\tilde{\boldsymbol{x}}_t^i). \tag{16}$$

In summary, the smoothing distribution, $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T})$, can be obtained using the MCP filter from time T to time 1 if the smoothing distribution, $p(\boldsymbol{x}_T|\boldsymbol{y}_{1:T})$, and a backward equation is given. Note that the smoothing distribution, $p(\boldsymbol{x}_T|\boldsymbol{y}_{1:T})$, can be obtained using the MCP filter (Algorithm 1). The algorithm of our backward equation smoothing is shown as Algorithm 2.

```
Algorithm 2: Backward Equation Smoothing
[{x̂̃^i_{t-1}}^M_{i=1}, llk] = Smoothing[{x̂̃^i_t}^M_{i=1}, y_{t-1}]
{
  FOR i=1,...M
    Predict: x^i_{t-1} ~ p(x̃_{t-1}|x̂̃^i_t, v^i_t)
    Weight: w̃^i_{t-1} = r(ψ(y_{t-1}, x̃^i_{t-1})) |∂ψ/∂y_{t-1}|
  ENDFOR
  Sum of Weights: sw = Σ_{i=1}^M w̃^i_{t-1}
  Log Likelihood: llk = log(sw/M)
  FOR i=1,...,M
```

Algorithm 2: Backward Equation Smoothing

$[\{\hat{\tilde{\boldsymbol{x}}}_{t-1}^i\}_{i=1}^M$, llk] = Smoothing$[\{\hat{\tilde{\boldsymbol{x}}}_t^i\}_{i=1}^M,\ \boldsymbol{y}_{t-1}]$

{

FOR i=1,...M

Predict: $\boldsymbol{x}_{t-1}^i \sim p(\tilde{\boldsymbol{x}}_{t-1}|\hat{\tilde{\boldsymbol{x}}}_t^i, v_t^i)$

Weight: $\tilde{w}_{t-1}^i = r(\psi(\boldsymbol{y}_{t-1}, \tilde{\boldsymbol{x}}_{t-1}^i))\left|\frac{\partial\psi}{\partial y_{t-1}}\right|$

ENDFOR

Sum of Weights: sw = $\sum_{i=1}^M \tilde{w}_{t-1}^i$

Log Likelihood: $llk = log(sw/M)$

FOR i=1,...,M

Normalize: $\tilde{w}_{t-1}^{i} = \frac{\tilde{w}_{t-1}^{i}}{sw}$

ENDFOR

Resampling: $[\{\hat{\tilde{\boldsymbol{x}}}_{t-1}^{i}\}_{i=1}^{M}]$ =resample$[\{\tilde{\boldsymbol{x}}_{t-1}^{i},\ \tilde{w}_{t-1}^{i}\}_{i=1}^{M}]$

RETURN$[\{\hat{\tilde{\boldsymbol{x}}}_{t-1}^{i}\}_{i=1}^{M}$, llk]

}

SmoothingMain $[\{\hat{\tilde{\boldsymbol{x}}}_{T}^{i}\}_{i=1}^{M},\ \{\boldsymbol{y}_t\}_{t=1}^{T}]$

{

Initialize: $llk = 0$

FOR t=T,...,2

smo = Smoothing$[\{\hat{\tilde{\boldsymbol{x}}}_{t}^{i}\}_{i=1}^{M},\ \boldsymbol{y}_{t-1}]$

$llk = llk + (llk\ in$ smo)

$\{\hat{\tilde{\boldsymbol{x}}}_{t-1}^{i}\}_{i=1}^{M} = (\{\hat{\tilde{\boldsymbol{x}}}_{t-1}^{i}\}_{i=1}^{M}\ in$ smo)

ENDFOR

RETURN$[\{\{\hat{\tilde{\boldsymbol{x}}}_{t}^{i}\}_{i=1}^{M}\}_{t=1}^{T}, llk]$

}

The computational complexity for our backward equation smoothing algorithm is $O(MT)$. It is equivalent to the computational complexity for the MCP filter and Godsill et al. (2004)[8]. Furthermore, our smoothing algorithm requires $O(M)$ storage to save particle weights because it requires only $w_T^i,\ \{1,\cdots,M\}$. By contrast, Godsill et al. (2004) requires $O(MT)$ storage to save weights of particle because it requires $w_t^i,\ \{1,\cdots,M\},\ \{1,\cdots,T\}$. In other words, the advantages of our smoothing algorithm are its simplicity and small memory requirement. However, there remains an obstacle. In general, the backward transition equation of states cannot be obtained analytically.

We propose a filter initialization algorithm, which chooses an appropriate initial distribution (a prior distribution) $\hat{p}(\boldsymbol{x}_0)$ of the MCP filter. In general, the initial distribution $\hat{p}(\boldsymbol{x}_0)$ of the MCP filter is unknown. State estimation based on the MCP filter and smoothing is improved if one can choose an appropriate initial probability. We propose a filter initialization algorithm based on the smoothing distribution $p(\boldsymbol{x}_1|\boldsymbol{y}_{1:T})$ and a backward

[8]Computational complexities of the two-filter formula (Kitagawa (1996)) and the maximum $a\ posteriori$ sequence estimation (Godsill et al. (2001)) are $O(M^2T)$.

equation. We can obtain an appropriate initial probability $\hat{p}(\boldsymbol{x}_0)$ as follows.

$$\begin{aligned}\hat{p}(\boldsymbol{x}_0) &\simeq \int p(\boldsymbol{x}_0|\boldsymbol{x}_1)p(\boldsymbol{x}_1|\boldsymbol{y}_{1:T})d\boldsymbol{x}_1 \\ &\simeq \frac{1}{M}\sum_{i=1}^{M} p(\boldsymbol{x}_0|\hat{\tilde{\boldsymbol{x}}}_1^i) \\ &\simeq \frac{1}{M}\sum_{i=1}^{M} \delta(\boldsymbol{x}_0 - \tilde{\boldsymbol{x}}_0^i).\end{aligned} \tag{17}$$

Furthermore, we propose the following steps to estimate the smoothing distribution $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:T}),\ \{t = 1, 2, \cdots, T\}$ using Algorithm 2 and Eq. (17) as follows.

1. Choosing an arbitrary initial distribution $p(\boldsymbol{x}_0)$.
2. Using Algorithms 1 and 2 with $p(\boldsymbol{x}_0)$.
3. Calculating the initial distribution $\hat{p}(\boldsymbol{x}_0)$ based on Eq. (17).
4. Using Algorithms 1 and 2 with $\hat{p}(\boldsymbol{x}_0)$.

3 Examples

We apply our algorithms to a linear Gaussian state space model, a linear non-Gaussian state space model, the stochastic volatility model, and a stochastic volatility model with a t-distribution. In the following subsections: (1) we generate an artificial time series ($T = 100$) based on each model ($x_0 \sim N(0, 1^2)$), (2) we estimate a state x_t using our backward smoothing algorithm[9], and (3) we calculate $\hat{p}(x_0)$. We set the number of particles, M, to 10000 to estimate the state.

3.1 Linear Gaussian State Space Model

A linear Gaussian state space model is defined as

$$\begin{aligned}x_t &= x_{t-1} + v_t, \\ y_t &= x_t + \epsilon_t,\end{aligned} \tag{18}$$

where $v_t \sim N(0, \sigma_s^2)$ and $\epsilon_t \sim N(0, \sigma_m^2)$. We set $\{\sigma_s, \sigma_m\} = \{1, 3\}$. The backward equation of Eq. (18) is given by

$$x_{t-1} = x_t - v_t. \tag{19}$$

In Fig. 1, the thick black line represents the estimated state x_t based on the backward equation smoothing, the dotted line represents the estimated state x_t based on the MCP filter, and the thin black line represents the real

[9] Initial particles are sampled from $Uniform(10, 11)$.

state. Figure 1 shows that state estimation based on our backward smoothing algorithm is improved at time points close to the start of the series. The initial distribution $\hat{p}(\boldsymbol{x}_0)$ and the true initial distribution $p(\boldsymbol{x}_0)$ are shown in Fig.

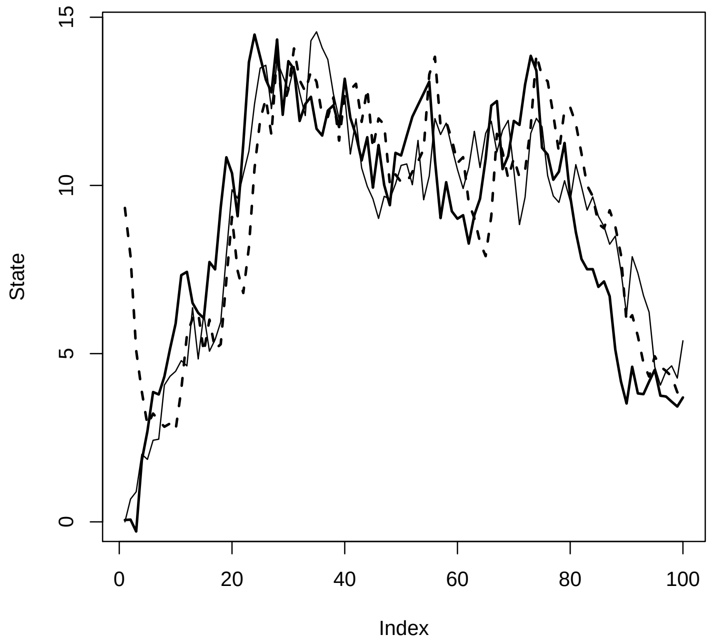

Figure 1: Linear Gaussian Model

2. It shows that $\hat{p}(\boldsymbol{x}_0)$ approximates $p(\boldsymbol{x}_0)$ well.

We compare backward smoothing with Kalman smoothing. In Figure 3, the thick black line represents the estimated state x_t based on the backward equation smoothing. The dotted line represents the estimated state x_t based on the Kalman filter; the thin black line represents the real state. Figure 3 shows that our algorithm realizes good state estimation as well as Kalman smoothing.

3.2 Linear Non-Gaussian State Space Model

In this subsection, we apply our algorithms to a simplest linear non-Gaussian state space model with a t-distribution. A simple linear non-Gaussian state space model with t-distribution is defined as

$$\begin{aligned} x_t &= x_{t-1} + v_t, \\ y_t &= x_t + \epsilon_t, \end{aligned} \tag{20}$$

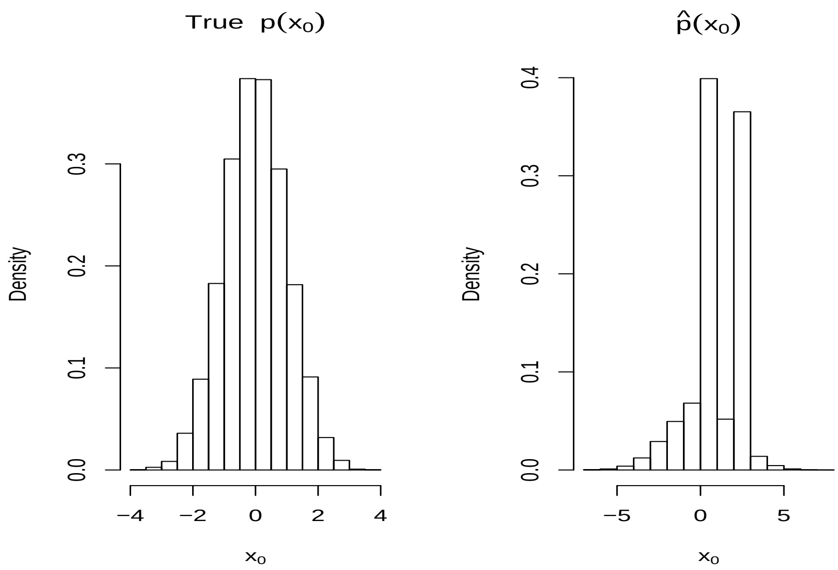

Figure 2: Initial Distribution (Linear Gaussian Model)

where $v_t \sim t(df)$[10] and $\epsilon_t \sim N(0, \sigma_m^2)$. We set $\{df, \sigma_m\} = \{8, 2\}$. The backward equation of Eq. (20) is given by

$$x_{t-1} = x_t - v_t. \tag{21}$$

The estimated state x_t is shown in Fig. 4. It shows that state estimation based on our backward smoothing is improved at time points close to the start of the series. The initial distribution $\hat{p}(\boldsymbol{x}_0)$ and the true initial distribution $p(\boldsymbol{x}_0)$ are shown in Fig. 5. It shows that $\hat{p}(\boldsymbol{x}_0)$ approximates $p(\boldsymbol{x}_0)$ well.

3.3 Stochastic Volatility Model

The stochastic volatility model, which is introduced by Taylor (1986), is adopted to model the autoregressive behavior of the volatility and non-Normality in the returns in financial time series. The simplest stochastic volatility model is defined as

$$\begin{aligned} x_t &= \alpha_s x_{t-1} + v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{22}$$

[10]The acronym df represents degrees of freedom of the t-distribution.

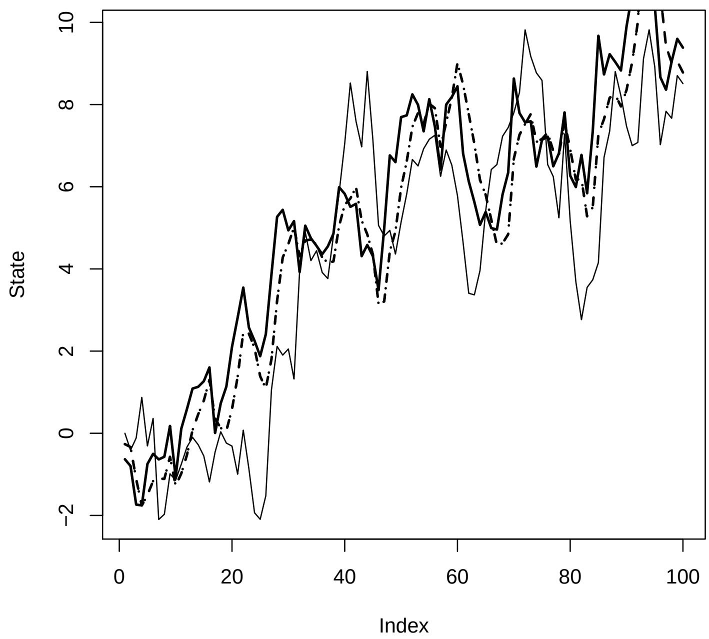

Figure 3: Compare Backward Smoothing with Kalman smoothing

where $v_t \sim N(0, \sigma_s^2)$, and $\epsilon_t \sim N(0, \sigma_m^2)$. We set $\{\alpha_s, \sigma_s, \sigma_m\} = \{0.8, 1, 1\}$. The backward equation of Eq. (22) is given by

$$x_{t-1} = \frac{1}{\alpha_s}(x_t - v_t). \tag{23}$$

The estimated state x_t is shown in Fig. 6. It shows that state estimation based on our backward smoothing is improved at time points close to the start of the series. The initial distribution $\hat{p}(\boldsymbol{x}_0)$ and the true initial distribution $p(\boldsymbol{x}_0)$ are shown in Fig. 7. It shows that $\hat{p}(\boldsymbol{x}_0)$ approximates $p(\boldsymbol{x}_0)$ well.

3.4 Stochastic Volatility Model with t-distribution

A stochastic volatility model with a t-distribution, which is introduced by Liesenfeld and Jung (2000), is defined as

$$\begin{aligned} x_t &= \alpha_s x_{t-1} + v_t, \\ y_t &= \epsilon_t \exp(\frac{x_t}{2}), \end{aligned} \tag{24}$$

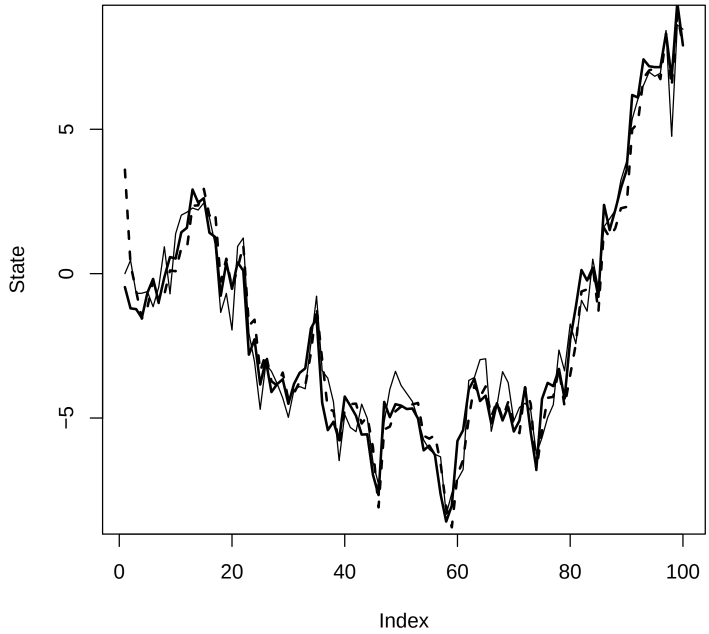

Figure 4: Linear Non-Gaussian Model

where $v_t \sim N(0, \sigma_s^2)$ and $\epsilon_t \sim t(df)$. We set $\{\alpha_s, \sigma_s, df\} = \{0.8, 1, 4\}$. The backward equation of Eq. (24) is given by

$$x_{t-1} = \frac{1}{\alpha_s}(x_t - v_t). \tag{25}$$

The estimated state x_t is shown in Fig. 8. It shows that state estimation based on our backward smoothing is improved at time points close to the start of the series. The initial distribution $\hat{p}(\boldsymbol{x}_0)$ and the true initial distribution $p(\boldsymbol{x}_0)$ are shown in Fig. 9. It shows that $\hat{p}(\boldsymbol{x}_0)$ approximates $p(\boldsymbol{x}_0)$ well.

4 Conclusions

We proposed a smoothing algorithm based on the Monte Carlo Particle filter and a backward equation. Our method is applicable to any nonlinear non-Gaussian state space model if a backward equation is obtained analytically. The advantage of our backward smoothing is its simplicity. It is a minor modification of the "standard" MCP filter. Moreover, our algorithm requires little memory to store the weights of particles. Nevertheless, an obstacle to its implementation remains: in general, the backward transition equation of states cannot be obtained analytically. Moreover, we propose a filter initialization algorithm based on the smoothing distribution $p(\boldsymbol{x}_1|\boldsymbol{y}_{1:T})$ and a back-

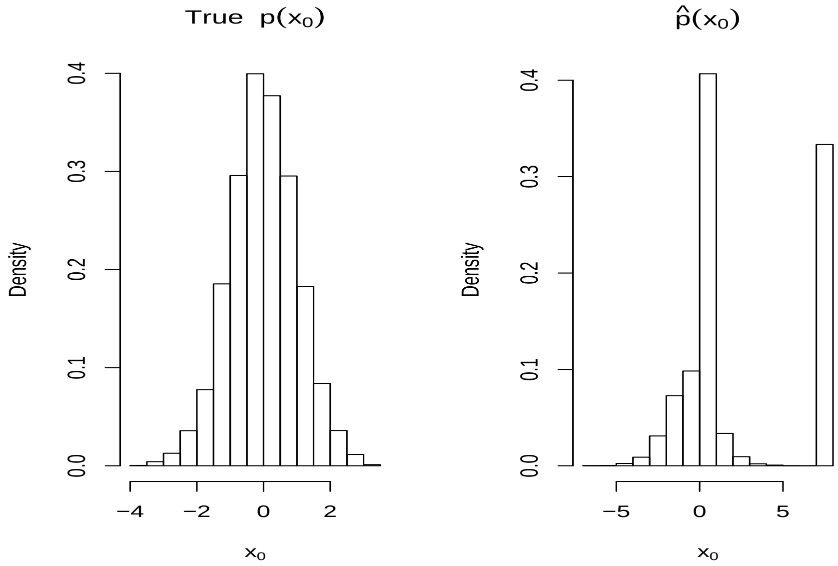

Figure 5: Initial Distribution (Linear Non-Gaussian Model)

ward equation. Our filter initialization algorithm is very simple to implement and realizes good approximation of a real initial distribution. We demonstrate the effectiveness of our method by applying it to a linear Gaussian state space model, a linear non-Gaussian state space model, a stochastic volatility model, and a stochastic volatility model with a t-distribution.

References

Arulampalam, M., S. Maskell, N. Gordon, and T. Clapp (2002) ‘A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking.’ *IEEE Transactions on Signal Processing* 50, 174–188

Briers, M., A. Doucet, and S. R. Maskell (2004) ‘Smoothing algorithms for state-space models.’ *Technical Report CUED/F-INFENG/TR.498.* Cambridge University Engineering Department

Doucet, A., N. de Freitas, and N. Gordon, eds (2001) *Sequential Monte Carlo Methods in Practice* (New York: Springer-Verlag)

Doucet, A., S. Godsill, and C. Andrieu (2000) ‘On sequential monte carlo sampling methods for Bayesian filtering.’ *Statistics and Computing* 10, 197–208

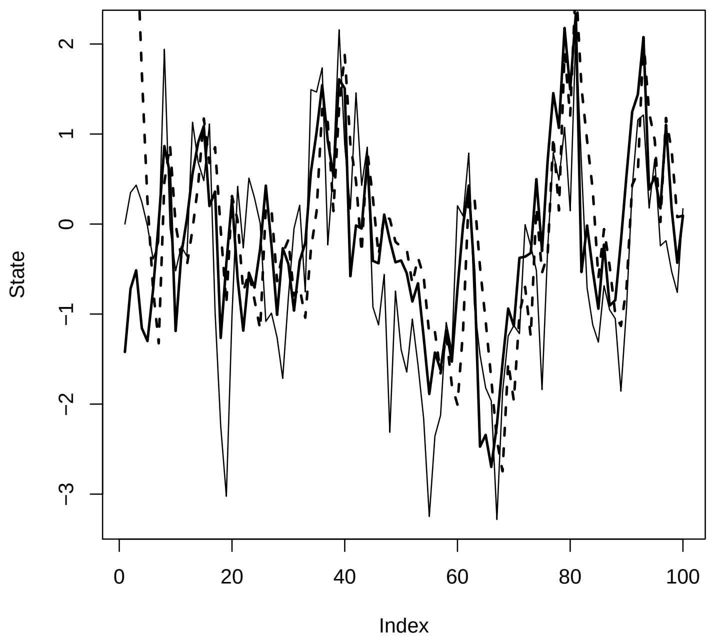

Figure 6: Stochastic Volatility Model

Godsill, S. J., A. Doucet, and M. West (2001) 'Maximum a posteriori sequence estimation using Monte Carlo particle filters.' *Annals of The Institute of Statistical Mathematics* 53, 82–96

— (2004) 'Monte Carlo smoothing for non-linear time series.' *Journal of the American Statistical Association* 99, 156–168

Gordon, N., D. Salmond, and A. Smith (1993) 'Novel approach to nonlinear/non-Gaussian Bayesian state estimation.' *IEEE Proceedings-F* 140, 107–113

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

Klaas, Mike, Mark Briers, Nando de Freitas, Arnaud Doucet, Simon Maskell, and Dustin Lang (2006) 'Fast particle smoothing: if I had a million particles.' *ACM International Conference Proceeding Series* 148, 481–488

Liesenfeld, R., and R. C. Jung (2000) 'Stochastic volatility models: Conditional normality versus heavy-tailed distributions.' *Journal of Applied Econometrics* 15, 137–160

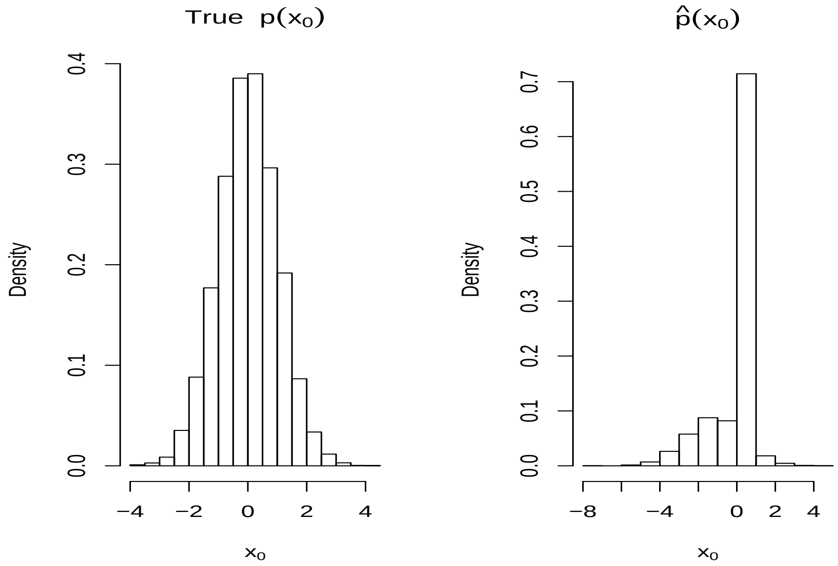

Figure 7: Initial Distribution (Stochastic Volatility Model)

Nakamura, K., and T. Tsuchiya (2007) 'A recursive recomputation approach for smoothing in nonlinear state space modeling – an attempt for reducing space complexity –.' *IEEE Transactions on Signal Processing.* Forthcoming

Taylor, S. J. (1986) *Modelling Financial Time Series* (Chichester: John Wiley)

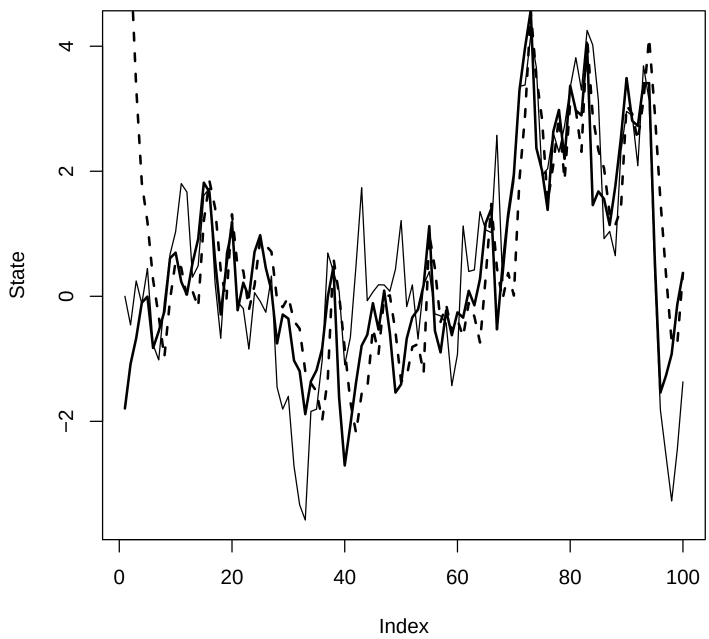

Figure 8: Stochastic Volatility Model with t-distribution

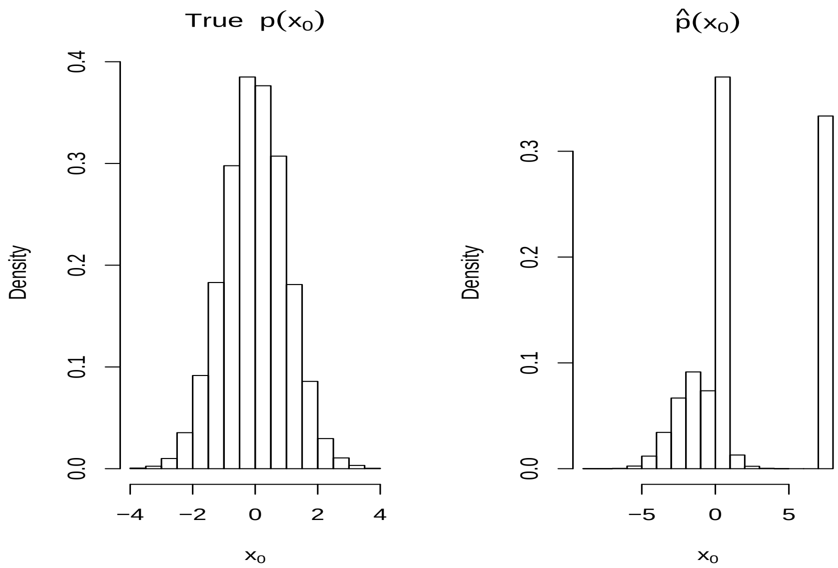

Figure 9: Initial Distribution (Stochastic Volatility Model with t-distribution)