UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Universidade Federal do Espirito Santo

A new method for sequential learning of states and Parameters for state space models. The particle swarm learning optimization.

Prof.Dr.Ivan R. Enriquez Guzman

Universidade Federal do Espirito Santo
DEST-CCE-UFES

8 de outubro de 2021

Summary

Definition: A sequence of evolving probability distributions $\pi_t(X_t)$ indexed by discrete time $t = 0, 1, \ldots,$ is called a *probabilistic dynamic system*, and the state variable X_t can evolve in the following tree ways

- increasing dimension: $X_{t+1} = (X_t, x_{t+1})$.
- discharging: X_{t+1} has one a fewer component than X_t
- no change: $X_{t+1} = X_t$

Note that, $\pi_t()$ is a target distribution of the D.S. The difference between π_t and π_{t+1} is because exit a new state. Usually we need estimate the $\pi_t(x_{t+1})$ (prediction), $\pi_{t+1}(x_t)$ (updating/smoothing) and $\pi_{t+1}(x_{t+1})$ (new estimation/ filtering).

The state-space model to Dynamic models

Dynamic models (DM) in state space models (SSM) were developed in control system, and have been implemented in several areas such as space engineering, signals processing, economy, environment and DNA sequence analysis. These models are described in two equations, the first referred to observations and the second referred to unknown states. Let θ and $\mathbf{h_t}$ be the vector of parameters and the set of states, a DM as follows:

$$y_t|\mathbf{h_t}, \theta \sim f(\ldots|\mathbf{h_t}, \theta), \tag{1a}$$

$$\mathbf{h_{t+1}}|\mathbf{h_t}, \theta \sim g(\mathbf{h_{t+1}}|\mathbf{h_t}, \theta). \tag{1b}$$

State-space models aim to fit dynamic models for sequentially observed data y_t. In these models, the states (h_t) and fixed parameters, are assumed to drive the data. State-space models are specified at each time t by the observation equation $f(y_t|\mathbf{h_t}, \boldsymbol{\theta})$ and the evolution equation $g(\mathbf{h_{t+1}}|\mathbf{h_t}, \boldsymbol{\theta})$.

The space state models to Dynamic Models

Let $\boldsymbol{\theta}$ and $\{h^i_{1:T}\}$ be the vector of parameters and the set of states, the model consists of two equations:

$$y_t = f(y_{t-1}, h_t, \boldsymbol{\theta}) + \varepsilon_t, \qquad t = 1, 2 \ldots, T, \tag{2a}$$

$$h_{t+1} = g(h_t, \boldsymbol{\theta}) + \eta_t, \qquad t = 1, 2 \ldots, T-1, \tag{2b}$$

We are interested in the posterior density $p(\boldsymbol{\theta}|y^t, h_{1:T})$, obtained from the joint distribution $p(\boldsymbol{\theta}, h_{1:T}|y^t)$, where y^t are the observations $y^t = \{y_1, \ldots, y_T\}$, i.e, the estimated sequential parameters and the state filter problem are associated to the joint posterior distribution, $p(h_t, \boldsymbol{\theta}|y_t)$.

The main problem in dynamic models:

The main problem in dynamic models:

"*The estimation of the parameters and states and* $\pi_t(\theta, x_t)$"

The main problem in dynamic models:

"*The estimation of the parameters and states and* $\pi_t(\theta, x_t)$"

Some methods:

The main problem in dynamic models:

"*The estimation of the parameters and states and* $\pi_t(\theta, x_t)$"

Some methods:

- Kalman filter method. (parametric method)

The main problem in dynamic models:

"*The estimation of the parameters and states and* $\pi_t(\theta, x_t)$"

Some methods:

- Kalman filter method. (parametric method)
  - f and g (2a-2b)are linear and Gaussian noise.(GL)

The main problem in dynamic models:

"*The estimation of the parameters and states and* $\pi_t(\theta, x_t)$"

Some methods:

- Kalman filter method. (parametric method)
  - f and g (2a-2b)are linear and Gaussian noise.(GL)
- For Non Gaussian and non-linear models:

The main problem in dynamic models:

"*The estimation of the parameters and states and* $\pi_t(\theta, x_t)$"

Some methods:

- Kalman filter method. (parametric method)
  - f and g (2a-2b)are linear and Gaussian noise.(GL)
- For Non Gaussian and non-linear models:
  - Extended Kalman Filter,**(author?)** (Sorenson e Alspach 1971).
  - Very slow, change the posterior function for Taylor approach and may lead to biased estimates,

The main problem in dynamic models:

"*The estimation of the parameters and states and* $\pi_t(\theta, x_t)$"

Some methods:

- Kalman filter method. (parametric method)
  - f and g (2a-2b)are linear and Gaussian noise.(GL)
- For Non Gaussian and non-linear models:
  - Extended Kalman Filter,**(author?)** (Sorenson e Alspach 1971).
  - Very slow, change the posterior function for Taylor approach and may lead to biased estimates,
- Numeric Monte Carlo method:
  We need:samples either drawn from $\pi_t(\theta, x_t)$, or drawn from another distribution $g_t()$ and weighted properly.
  - MC plus MCMC schemes (to Static methods) treat each π_t separately and repeating the same kind of iterative processes. All draws obtained at time t are discarded in the evolve from π_t to π_{t+1}.

The main problem in dynamic models:

"*The estimation of the parameters and states and* $\pi_t(\theta, x_t)$"

Some methods:

- Kalman filter method. (parametric method)
  - f and g (2a-2b)are linear and Gaussian noise.(GL)
- For Non Gaussian and non-linear models:
  - Extended Kalman Filter,**(author?)** (Sorenson e Alspach 1971).
  - Very slow, change the posterior function for Taylor approach and may lead to biased estimates,
- Numeric Monte Carlo method:
  We need:samples either drawn from $\pi_t(\theta, x_t)$, or drawn from another distribution $g_t()$ and weighted properly.
  - MC plus MCMC schemes (to Static methods) treat each π_t separately and repeating the same kind of iterative processes. All draws obtained at time t are discarded in the evolve from π_t to π_{t+1}.
  - Sequence Monte Carlo SMC (Particle Filters PF):
    - Fast method but may lead to biased estimates,

Some problems, for these methods:

Some problems, for these methods:

- In all SISR (IS,SIS,SIR) procedures, the discrete representation of $g(h_t)$ by a sample h_t^j with weight w_t^j degenerates very rapidly as the number of resampling increases. Then, estimating $E(H(h_t))$ can be very inaccurate.

Some problems, for these methods:

- In all SISR (IS,SIS,SIR) procedures, the discrete representation of $g(h_t)$ by a sample h_t^j with weight w_t^j degenerates very rapidly as the number of resampling increases. Then, estimating $E(H(h_t))$ can be very inaccurate.
- IS allows us to sample from complex highly-dimensional distributions though exhibits linear increases in complexity upon each subsequent draw (Doucet and Johansen 2008).
- SIS method is sensitive to the choice of the proposed density, (state transition probability). The variance of estimates increases exponentially with n and leads to fewer and fewer non-zero importance weights (Doucet and Johansen 2008). This problem is known as weight degeneracy.
- SIR leads to a lack of diversity amongst particles, a problem is known as sample impoverishment (there are many repeated points). Thus the weight degeneracy and sample impoverishment are part of one larger correlated problem.

For the parameters:

Several methods to Dynamic Models

For the parameters:

- MCMC methods plus Kalman filter.
  - using approximation/linearlization (Taylor approach),

Several methods to Dynamic Models

For the parameters:

- MCMC methods plus Kalman filter.
  - using approximation/linearlization (Taylor approach),
  - computationally intensive, and slow convergence.

Several methods to Dynamic Models

For the parameters:

- MCMC methods plus Kalman filter.
  - using approximation/linearlization (Taylor approach),
  - computationally intensive, and slow convergence.
- Sequential Monte Carlo or Particle Filters

Several methods to Dynamic Models

For the parameters:

- MCMC methods plus Kalman filter.
  - using approximation/linearlization (Taylor approach),
  - computationally intensive, and slow convergence.
- Sequential Monte Carlo or Particle Filters
  - Bootstrap Filter,Auxiliar Particle Filter ((Shephard e Pitt 1997)). Use importance sampling.

Several methods to Dynamic Models

For the parameters:

- MCMC methods plus Kalman filter.
  - using approximation/linearlization (Taylor approach),
  - computationally intensive, and slow convergence.
- Sequential Monte Carlo or Particle Filters
  - Bootstrap Filter,Auxiliar Particle Filter ((Shephard e Pitt 1997)). Use importance sampling.
- Particle filter/learning Liu and West

Several methods to Dynamic Models

For the parameters:

- MCMC methods plus Kalman filter.
  - using approximation/linearlization (Taylor approach),
  - computationally intensive, and slow convergence.
- Sequential Monte Carlo or Particle Filters
  - Bootstrap Filter,Auxiliar Particle Filter ((Shephard e Pitt 1997)). Use importance sampling.
- Particle filter/learning Liu and West
  - Kernel smoothing in Mixture Gaussian Dist.
  - Use SIR for the states.
  - result biased estimates in NLNG models.

Summary:

Summary:

- Many paper highlight the potential inaccuracy of SIS estimation due to the large variance of the importance weights.

Summary:

- Many paper highlight the potential inaccuracy of SIS estimation due to the large variance of the importance weights.
- Most particles continue being used despite having a negligible weight, which causes accuracy and performance problems. This is know as weight degeneracy problem and consists in the approximation to zero of most weights after a few iterations.

Summary:

- Many paper highlight the potential inaccuracy of SIS estimation due to the large variance of the importance weights.
- Most particles continue being used despite having a negligible weight, which causes accuracy and performance problems. This is know as weight degeneracy problem and consists in the approximation to zero of most weights after a few iterations.
- The Sequential Importance Re-sampling (SIR) attempts to solve the problem, by replacing high weight particles with many particles with high correlation among them.
- Re-sampling, however, leads to the marginal distribution of the state collapsing onto a few or single unique particles, thus eventually leading to a problem similar to weight degeneracy.

Summary:

- Many paper highlight the potential inaccuracy of SIS estimation due to the large variance of the importance weights.
- Most particles continue being used despite having a negligible weight, which causes accuracy and performance problems. This is know as weight degeneracy problem and consists in the approximation to zero of most weights after a few iterations.
- The Sequential Importance Re-sampling (SIR) attempts to solve the problem, by replacing high weight particles with many particles with high correlation among them.
- Re-sampling, however, leads to the marginal distribution of the state collapsing onto a few or single unique particles, thus eventually leading to a problem similar to weight degeneracy.

Liu and West,2001, suggested the use of kernel smoothing in mixture multivariate Gaussian distributions to approximate the posterior distribution for the parameters of the model, to overcome the sampling problem of the parameter.

Solution propose

- The analysis of the aforementioned works led us to identify that the process of sampling for a generation of parameter values (one each time) conditioned to few states causes the problem of the convergence.
- We claim that by evaluating several parameter values simultaneously we avoid the degeneration problem.
- We propose the adoption of the Bare Borne Particle Swarm Optimization (BBPSO) method for sequential learning of states and parameters.

The idea of analyzing multiple values at a time is complemented with information exchange among them, moves us to propose the use of the Particle Swarm Optimization (PSO) method.

Filtering for States and Parameter

In the general for fixed parameters model θ, considering the joint distribution,

- the sample-based framework starting at the time t, we obtain a joint sample $h_t^j, \theta_t^j : j = 1, \ldots, N$ with weights w_t^j.
- Based on y_{t+1}, we generate a sample from $p(h_{t+1}, \theta | F_{t+1})$.

By Bayes theorem we have

$$p(\mathbf{h}_{t+1}, \boldsymbol{\theta} | F_{t+1}) \propto p(y_{t+1} | \mathbf{h}_{t+1}, \boldsymbol{\theta}) p(\mathbf{h}_{t+1}, \boldsymbol{\theta} | F_t) \quad (3a)$$

$$\propto p(y_{t+1} | \mathbf{h}_{t+1}, \boldsymbol{\theta}) p(\mathbf{h}_{t+1} | \boldsymbol{\theta}, F_t) p(\boldsymbol{\theta} | F_t) \quad (3b)$$

- this is, density function $p(\theta | F_t)$ is an important ingredient in the update.

- Gordon (1): to reducing the sample degeneracy/attrition problem :($\theta_{t+1} = \theta_t + \epsilon_{t+1}$ and $\epsilon_t \sim N(0, W_{t+1})$ where θ_t and ϵ_{t+1} are independent.
- In literature, were suggested many others variations of this method.
- **(author?)** (Liu e West 2001) suggested a kernel smoothing approximation of $p(\theta|y_t)$, by a mixture of multivariate normals **(author?)** (West e Harrison 1997).

(author?) (Liu e West 2001) suggested a kernel smoothing approximation of $p(\theta|y_t)$ by a mixture of multivariate normals. Let be $\{\boldsymbol{h}_t^{(l)}, \boldsymbol{\theta}_t^{(l)}, \boldsymbol{w}_t^{(l)}\}_{l=1}^N \sim \hat{p}(h_t, \boldsymbol{\theta}|y_t)$ a sample, then the posterior approximation of the parameter is given by $\hat{p}(\boldsymbol{\theta}|Y_{1:t}) = \sum_{j=1}^N w_t^j \boldsymbol{N}(a\boldsymbol{\theta}_t^{(j)} + (1-a)\hat{\boldsymbol{\theta}}_t; h^2 \boldsymbol{V}_t)$

$$\begin{aligned} \hat{\boldsymbol{\theta}}_t &= \sum_{j=1}^N w_t^{(j)} \theta_t^{(j)}, \\ \boldsymbol{V}_t &= \sum_{j=1}^N w_t^{(j)} (\theta_t^{(j)} - \bar{\theta}_t)(\theta_t^{(j)} - \bar{\theta}_t)^T, \end{aligned}$$

a and h are function of $\delta \in (0, 1]$, with $h^2 = 1 - ((3\delta - 1)/2\delta)^2$ and $a = \sqrt{1 - h^2}$. **(author?)** (Liu e West 2001) suggested δ around 0.95-0.99.

Algorithm 1 (Liu and West)

Algorithm 1

1. k^l is sampled from $p(y_{t+1}|\mu_{t+1}^{(j)}, \boldsymbol{\theta}_t^{(j)})w_t^{(j)}$ with $\mu_{t+1}^{(j)} = E(h_{t+1}|h_t, \boldsymbol{\theta})$
2. $\boldsymbol{\theta}_{t+1}^{(l)}$ is sampled from $\boldsymbol{N}(a\boldsymbol{\theta}_t^{(k^l)} + (1-a)\hat{\boldsymbol{\theta}}_t; h^2\boldsymbol{V}_t)$
3. $\boldsymbol{h}_{t+1}^{(l)}$ is sampled from $p(h_{t+1}|h_t^{(k^l)}, \boldsymbol{\theta}_{t+1}^{(l)})$ which leads to weights $\boldsymbol{w}_{t+1}^{(l)} \propto \frac{\boldsymbol{p}(y_{t+1}|h_{t+1}^{(l)}, \boldsymbol{\theta}_{t+1}^{(l)})}{\boldsymbol{p}(y_{t+1}|\mu_{t+1}^{k^{(l)}}, m_t^{k^l})}$
4. Sample from the posterior, $\{\boldsymbol{h}_{t+1}^{(l)}, \boldsymbol{\theta}_{t+1}^{(l)}, \boldsymbol{w}_{\boldsymbol{t+1}}^{(\boldsymbol{l})}\}_{l=1}^N \sim p(h_{t+1}, \boldsymbol{\theta}|y_{t+1})$.

Liu and West have generalized the APF method, to adapt sequential parameter learning.This method, though, still has the problem of producing degenerate parameter estimates, which complicates accurate forecasting.

Bare Borne Particle Swarm Optimization (BBPSO)

Let $\mathbb{S} \subset \mathbb{R}^D$ be the search space of an objective function f.

- BBPSO is an algorithm based on a population living in $\mathbb{S}$.
- The population is referred to as swarm and its individuals are referred to as particles.
- The position of a particle in $\mathbb{S}$ represents a potential solution,
- Each particle keeps the memory of the personal best solution found during the search process
- The particles use a *neighborhood system* to exchange information between them. So, each particle also keeps the memory of the best solution found by any particle in its neighborhood.

Thus, the particles are influenced by their own previous experiences and by the experiences of their neighbors.

Swarms of Birds and bees - Particle Swarms Optimization.

Bare Bones Particle Swarm Optimization in particle filters

- Consider a swarm $\mathcal{S}$ with K particles. The position of a particle is denoted by an D-dimensional vector $\boldsymbol{x}_k^\tau = (x_{k1}^\tau, \ldots, x_{kD}^\tau)'$ in $\mathbb{S}$.
- The index k $(k = 1, \ldots, K)$ labels the kth particle in $\mathcal{S}$ and the index τ $(\tau = 1, 2, \ldots)$ represents the iteration,
- The neighborhood of a particle is a set $\mathcal{N}_k$ of particles which is able to communicate with $(\mathcal{N}_k \subseteq \mathcal{S})$
- The personal and neighborhood best positions are respectively denoted by $\boldsymbol{p}_k^\tau$ and $\boldsymbol{n}_k^\tau$.

Bare Bones Particle Swarm Optimization in particle filters

The swarm is initialized with random positions in $\mathbb{S}$. On the initialization, $\boldsymbol{p}_k^1 = \boldsymbol{x}_k^1$ for all k and $\boldsymbol{n}_k^1$ is given by:

$$\boldsymbol{n}_k^1 = \text{BEST}(\boldsymbol{p}_l^1 | l \in \mathcal{N}_k) = \arg\max\{f(\boldsymbol{p}_l^1) | l \in \mathcal{N}_k\} \tag{4}$$

Bare Bones Particle Swarm Optimization in particle filters

The swarm is initialized with random positions in $\mathbb{S}$. On the initialization, $\boldsymbol{p}_k^1 = \boldsymbol{x}_k^1$ for all k and $\boldsymbol{n}_k^1$ is given by:

$$\boldsymbol{n}_k^1 = \text{BEST}(\boldsymbol{p}_l^1 | l \in \mathcal{N}_k) = \arg\max\{f(\boldsymbol{p}_l^1) | l \in \mathcal{N}_k\} \tag{4}$$

The position of a particle is updated as follows:

$$\boldsymbol{x}_k^{\tau+1} = \boldsymbol{\mu}_k^\tau + \boldsymbol{\sigma}_k^\tau \odot \boldsymbol{z} \tag{5}$$

where

- $\boldsymbol{\mu}_k^\tau = (\mu_{k1}^\tau, \ldots, \mu_{kD}^\tau)' = \frac{1}{2}(p_{k1}^\tau + n_{k1}^\tau, \ldots, p_{kD}^\tau + n_{kD}^\tau)'$
- $\boldsymbol{\sigma}_k^\tau = (\sigma_{k1}^\tau, \ldots, \sigma_{kD}^\tau)' = (|p_{k1}^\tau - n_{k1}^\tau|, \ldots, |p_{kD}^\tau - n_{kD}^\tau|)'$
- $\boldsymbol{z} = (z_1, \ldots, z_D)' \sim (N_1(0,1), \overset{idd}{\ldots}, N_D(0,1))'$.

This is,

$$x_{kd}^{\tau+1} \sim N(\mu_{kd}^{\tau}, (\sigma_{kd}^{\tau})^2) \tag{6}$$

for all k, d and $\tau \geq 1$. In addition, $\boldsymbol{x}_k^{\tau+1}$ has multivariate normal distribution with mean vector $\boldsymbol{\mu}_k^{\tau}$ and covariance matrix $\boldsymbol{\Sigma}_k^{\tau}$, that is, $\boldsymbol{x}_k^{\tau+1} \sim \boldsymbol{N}(\boldsymbol{\mu}_k^{\tau}, \boldsymbol{\Sigma}_k^{\tau})$ where

$$\boldsymbol{\Sigma}_k^{\tau} = \text{diag}(|p_{k1}^{\tau} - n_{k1}^{\tau}|^2, \ldots, |p_{kD}^{\tau} - n_{kD}^{\tau}|^2).$$

This is,

$$x_{kd}^{\tau+1} \sim N(\mu_{kd}^{\tau}, (\sigma_{kd}^{\tau})^2) \tag{6}$$

for all k, d and $\tau \geq 1$. In addition, $\boldsymbol{x}_k^{\tau+1}$ has multivariate normal distribution with mean vector $\boldsymbol{\mu}_k^{\tau}$ and covariance matrix $\boldsymbol{\Sigma}_k^{\tau}$, that is, $\boldsymbol{x}_k^{\tau+1} \sim \boldsymbol{N}(\boldsymbol{\mu}_k^{\tau}, \boldsymbol{\Sigma}_k^{\tau})$ where

$$\boldsymbol{\Sigma}_k^{\tau} = \text{diag}(|p_{k1}^{\tau} - n_{k1}^{\tau}|^2, \ldots, |p_{kD}^{\tau} - n_{kD}^{\tau}|^2).$$

After updating the position, the personal best position is updated as follows:

$$\boldsymbol{p}_k^{\tau+1} = \text{BEST}(\boldsymbol{x}_k^{\tau+1}, \boldsymbol{p}_k^{\tau}). \tag{7}$$

This is,

$$x_{kd}^{\tau+1} \sim N(\mu_{kd}^{\tau}, (\sigma_{kd}^{\tau})^2) \tag{6}$$

for all k, d and $\tau \geq 1$. In addition, $\boldsymbol{x}_k^{\tau+1}$ has multivariate normal distribution with mean vector $\boldsymbol{\mu}_k^{\tau}$ and covariance matrix $\boldsymbol{\Sigma}_k^{\tau}$, that is, $\boldsymbol{x}_k^{\tau+1} \sim \boldsymbol{N}(\boldsymbol{\mu}_k^{\tau}, \boldsymbol{\Sigma}_k^{\tau})$ where

$$\boldsymbol{\Sigma}_k^{\tau} = \text{diag}(|p_{k1}^{\tau} - n_{k1}^{\tau}|^2, \ldots, |p_{kD}^{\tau} - n_{kD}^{\tau}|^2).$$

After updating the position, the personal best position is updated as follows:

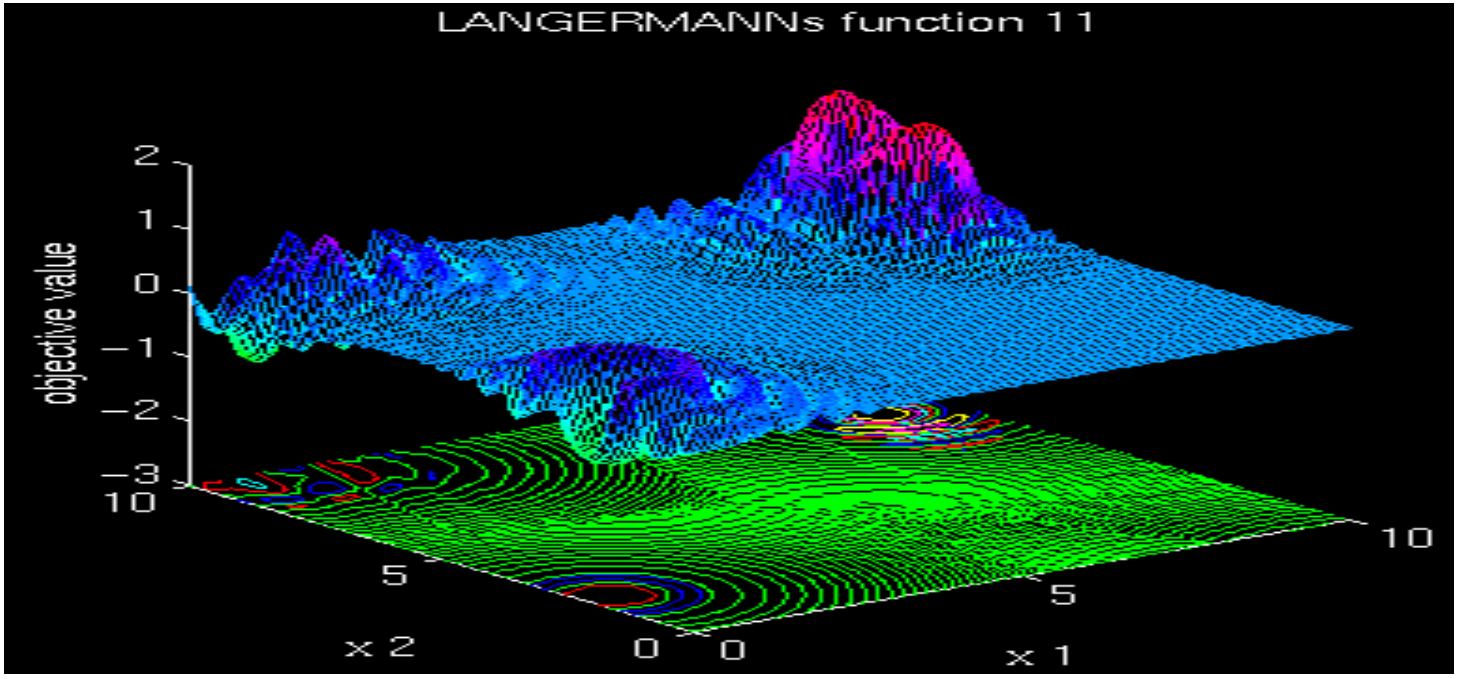
$$\boldsymbol{p}_k^{\tau+1} = \text{BEST}(\boldsymbol{x}_k^{\tau+1}, \boldsymbol{p}_k^{\tau}). \tag{7}$$

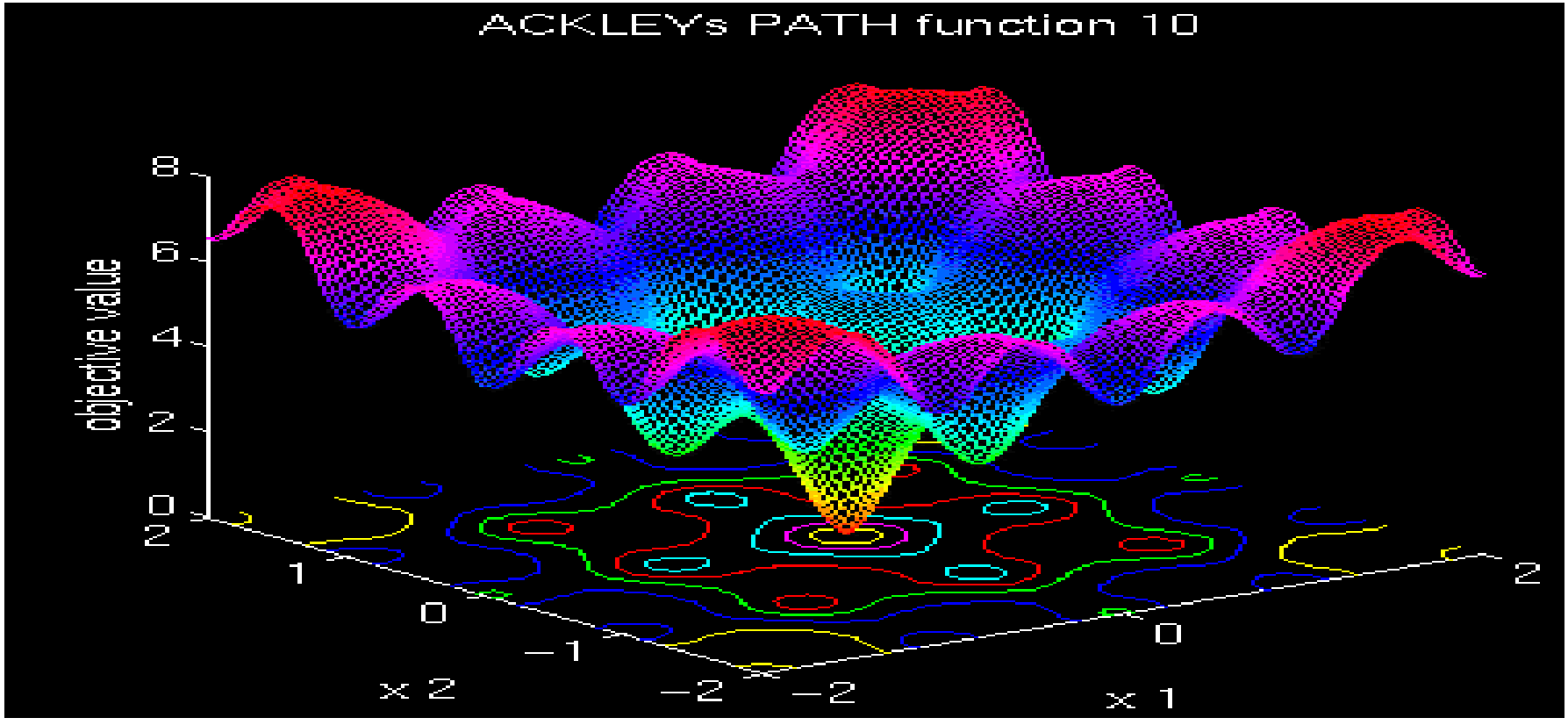
Finally, the neighborhood best position is given by:

$$\boldsymbol{n}_k^{\tau+1} = \text{BEST}(\boldsymbol{p}_l^{\tau+1} | l \in \mathcal{N}_k). \tag{8}$$

The essential steps of the canonical BBPSO can be summarized as the pseudo code shown in Algorithm 2.

1. Require $K, \mathcal{N}, \tau_{max}, \beta, f$, boderline : $\underline{x}, \bar{x}$.
2. To $\tau \leftarrow 1$
   - FOR $k \in \{1 \ldots K\}$ $\boldsymbol{x}_k \leftarrow \underline{\boldsymbol{x}} + (\bar{\boldsymbol{x}} - \underline{\boldsymbol{x}}) \odot \boldsymbol{U}(0,1)$ and $\boldsymbol{p}_k \leftarrow \boldsymbol{x}_k$, END FOR.
   - FOR $k \in \{1 \ldots K\}$ $\boldsymbol{n}_k \leftarrow$ BEST$(\boldsymbol{p}_l | l \in \mathcal{N}_k)$ END FOR,
3. Repeat $\tau \leftarrow \tau + 1$
   - FOR $k \in \{1 \ldots K\}$ $\boldsymbol{x}_k \leftarrow \boldsymbol{\mu}_k + \Sigma^{\frac{1}{2}} z$ according to Eq.(5) END FOR,
   - FOR $k \in \{1 \ldots K\}$ $\boldsymbol{p}_k^{\tau} \leftarrow$ BEST$(\boldsymbol{x}_k^{\tau}, \boldsymbol{p}_k^{\tau-1})$, END FOR,
   - FOR $k \in \{1 \ldots K\}$ $\boldsymbol{n}_k \leftarrow$ BEST$(\boldsymbol{p}_l | l \in \mathcal{N}_k)$, END FOR
4. UNTIL some termination condition is met,(For example $\tau = T$)
5. Do $\boldsymbol{g} \leftarrow$ BEST$(\boldsymbol{n}_k | k = 1, \ldots, K)$
6. RETURN $\boldsymbol{g}$ and $f(\boldsymbol{g})$


LANGERMANNs function 11
objective value
2
1
0
−1
−2
−3
10
5
0
x 2
0
5
10
x 1


ACKLEYs PATH function 10
objective value
8
6
4
2
0
2
1
0
−1
−2
x 2
−2
0
2
x 1

Particle Swarm Learning optimization (PSLO)

Proposed method:

The resulting general algorithm as follows,

- Resample $\left\{(h_t, \theta_t)^i\right\}_{i=1}^N \sim \left\{h_t^{'}, \theta_t\right\}_{i=1}^N$ with weights $w_t \propto p(y_t|g(h_{t-1}), m^{(i)}(\theta) = \hat{\theta}^i)$ and $g(h_{t-1}) = \alpha + \beta h_{t-1}$.
- Optimization $(\theta_{t+1}) = \hat{\theta}_{t+1} = arg.max(f(\theta|h_t))$ via BBPSO.
- Sampler $\{(h_{t+1})\}_{i=1}^N \sim p(h_{t+1}|\hat{\theta}_{t+1}, h_t)$, which leads to weights $w_{t+1}^l \propto \frac{p(y_{t+1}|h_{t+1}^l, \hat{\theta}_{t+1})}{p(y_{t+1}|g(h_t), \hat{\theta}_{t+1})}$,
- Sampler from the posterior, $\left\{h_{t+1}^l, \theta_{t+1}^l, w_{t+1}^l\right\}_{l=1}^N \propto p(h_{t+1}, \hat{\theta}_{t+1}|y^{t+1})$.
- Finaly, we find $\hat{\theta}^{Opt}$ via BBPSO, this is, $\hat{\theta}^{Opt} \leftarrow arg.max(f(\theta|h_{t+1})$ with $Tmax = 100$.
- optionalment we have the l previtions via BBPSO.

This method is summary in the next algorithm,

Particle Swarm Learning optimization (PSLO)

- Require the number of particles N (PF), $K, \mathcal{N}, c_1, c_2$ (BBPSO).
- Initialize, m_0, $C_0, \nu_0, \tau_0^2, b_0$ and B_0 in PF.
- For $t \in \{1, \ldots, n\}$
  - Resample $\left\{(h_t, \theta_t)^i\right\}_{i=1}^N \sim \left\{h_t^{`}, \theta_t\right\}_{i=1}^N$,
  - K^l is sampled with weights $w_t \propto p(y_t|\mu_t^j, \boldsymbol{\theta}_t^{(j)})$, $\mu_t^j = E(h_t|h_{t-1}, \boldsymbol{\theta})$;
- Estimation: via BBPSO. Initialize, $D_1, \mathbb{S}_1 \subset \mathbb{R}^{D_1}$, and τ_{MAX}^1.
  - $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{t+1} \leftarrow \text{argmax}(log(p(\theta|h_t, y_t)))$
- (Propagate) $\boldsymbol{h}_{t+1}^{(l)}$ is sampled from $p(h_{t+1}|h_t^{(k^l)}, \boldsymbol{\theta}_{t+1}^{(l)})$ with leads to weights $\boldsymbol{w}_{t+1}^{(l)} \propto \frac{\boldsymbol{p}(y_{t+1}|h_{t+1}^{(l)}, \boldsymbol{\theta}_{t+1}^{(l)})}{\boldsymbol{p}(y_{t+1}|\mu_{t+1}^{k(l)}, \boldsymbol{\theta}_t^{k^l})}$
- Sample from the posterior, $\{\boldsymbol{h}_{t+1}^{(l)}, \boldsymbol{\theta}_{t+1}^{(l)}, \boldsymbol{w}_{\boldsymbol{t+1}}^{(\boldsymbol{l})}\}_{l=1}^N \sim p(h_{t+1}, \boldsymbol{\theta}_{t+1}|y_{t+1})$.
- Finally, we find $\hat{\theta}^{Opt}$ via BBPSO, this is, $\hat{\theta}^{Opt} \leftarrow arg.max(f(\theta|h_{t+1})$ to compared with the estimation.
- ENDFOR
- RETURN $\hat{\boldsymbol{\theta}}$,$\mathbf{h}_{n+1:T}$ and $\mathbf{h}_{T+1:T+l}$.

Example: The basic Stochastic Volatility model

Among the variants of the stochastic volatility model, (Taylor 1982) proposed a SV model in discrete time as follows:

$$y_t = e^{\frac{h_t}{2}} \varepsilon_t, \qquad t = 1, 2 \ldots, T, \tag{9a}$$

$$h_{t+1} = \alpha + \phi h_t + \sigma_\eta \eta_t, \qquad t = 1, 2 \ldots, T-1, \tag{9b}$$

where y_t and h_t are the returns (corrected by the mean) and the log-volatility at time t respectively, which is assumed to follow a stationary process, $|\phi| < 1$ is the persistence in the volatility, σ_η is the standard deviation of log-volatility. The innovations ε_t and η_t are assumed independent, with standard Gaussian distribution, and $h_1 \sim N(\mu; \frac{\sigma_\eta^2}{(1-\phi^2)})$. We use the notation SV-N to denote this model and $\boldsymbol{\theta} = (\mu, \sigma_\eta^2, \phi)$ the parameter of the SV model.

We can express the likelihood functions and the distribution of the states as

$$p(\mathbf{y}_{1:T} \mid \mathbf{h}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{\theta}) \propto \prod_{t=1}^{T} \frac{\lambda_t^{1/2}}{(e^{h_t})^{1/2}} e^{-\frac{\lambda_t^{1/2}}{2e^{h_t}} y_t^2}, \tag{10a}$$

$$p(\mathbf{h}_{1:T} \mid \boldsymbol{\theta}) \propto \sqrt{\frac{(1-\phi^2)}{2\sigma_\eta^2}} \times e^{\frac{-(1-\phi^2)}{2\sigma_\eta^2}(h_1 - \frac{\alpha}{1-\phi})^2} \times \prod_{t=2}^{T} \frac{1}{\sigma_\eta} e^{\frac{-1}{2\sigma_\eta^2}(h_t - \alpha - \phi h_{t-1})^2}, \tag{10b}$$

$$p(\boldsymbol{\lambda}_{1:T}) \propto \prod_{t=1}^{T} p(\lambda_t / v) = \prod_{t=1}^{T} \frac{\frac{\nu}{2}^{\frac{\nu}{2}}}{\Gamma(\frac{\nu}{2})} \lambda_t^{\frac{\nu}{2}} e^{(\frac{-\nu}{2}\lambda_t)}. \tag{10c}$$

Therefore a joint distribution function $p(\boldsymbol{\theta}, \mathbf{h_t} | \mathbf{y_t})$ is given by the following product

$$L(\boldsymbol{\theta}, \mathbf{h}_{1:T} | \mathbf{y}_{1:T}) = p(\mathbf{y}_{1:T} \mid \mathbf{h}_{1:T}, \boldsymbol{\theta}) \times p(\mathbf{h}_{1:T} \mid \boldsymbol{\theta}) \times p(\boldsymbol{\lambda}_{1:T} \mid \nu), \tag{11}$$

where $\boldsymbol{\theta} = (\alpha, \phi, \sigma_\eta^2)$ and $\nu = 2, \ldots, 20$. For PSLO method, we use the Algorithm PSLO with $K = 30$ $\tau_{MAX} := 100$ iterations..

Note that, we have two maximization functions, for the parameters and forecasting the volatility. Accordingly, both objective functions are described as follows

$$f(\boldsymbol{\theta}) \propto log\frac{\sqrt{1-\phi^2}}{\sigma_\eta} - \frac{(1-\phi^2)}{2\sigma_\eta^2}(h_1 - \frac{\alpha}{1-\phi})^2 - \sum_{t=2}^{n}\frac{(h_t - \alpha - \phi h_{t-1})^2}{2\sigma_\eta^2} + log(\sigma_\eta^2), \quad (12a)$$

$$f(\mathbf{h}_t) = -\frac{1}{2}\sum_{t=1}^{T}\frac{(h_t - \alpha - \phi h_{t-1})^2}{\sigma_\eta^2} - \frac{T}{2}log\sigma_\eta^2. - \frac{1}{2}\sum_{t=1}^{T}(h_t + \frac{\lambda_t}{e^{h_t}}y_t^2), \quad (12b)$$

where h_t is the l-forecast states in particle learning and $K = l$ is the number of smooth states via BBPSO.

Fig. 1 shows the evolution of the parameters, which illustrates the learning process via PL-LW, delayed approximately 1000 iterations to find the stable process for the parameters. Since we considered the possible efficiency improvement, we decided to include the last of each 60 samples generated by the Liu and West methodology. Nonetheless, a high inefficiency factor remained in both cases (Table 1 and 2).
Figures 2 and 3 show the convergence process PSLO in both time series, we can see that for a few numbers of iterations (30 approx.) the sequence given by PSLO converge to the optimal value, meantime in PL sequences need 1000 iterations approximately.

	LW				LW-J			PSLO			
	α	ϕ	σ	time	α	ϕ	σ	α	ϕ	σ	time
No	336.32	330.43	281.54	76,13	7.12	7.05	17.44	11.32	4.07	4.33	43,58
T2	328.73	329.72	328.64	93,92	7.82	8.53	15.73	9.51	8.25	4.07	66,97
T3	302.95	322.91	330.92	92,64	9.29	9.27	7.90	10.23	11.64	13.53	66,54
T4	325.09	327.66	225.94	94,62	4.94	5.48	14.93	5.97	16.71	4.52	68,76
T5	235.90	340.77	223.24	102,12	15.53	15.74	16.18	11.95	16.28	6.65	67,8
T6	305.80	310.96	331.67	100,41	8.52	10.63	25.56	15.34	5.79	6.82	69,5
T7	331.30	332.48	352.20	101,46	6.25	6.84	16.58	10.58	14.14	12.46	66,86
T8	292.70	289.75	313.17	99,28	32.59	32.64	18.46	15.56	15.71	18.58	68,22
T9	323.41	323.50	254.7	97,98	16.90	20.14	22.99	7.85	14.64	12.37	68,81
T10	316.56	326.02	317.34	100,34	6.75	6.78	17.55	17.67	9.75	12.21	67,6
T11	327.45	325.90	249.60	100,54	17.33	18.13	25.82	9.27	9.77	7.85	69,64
T12	265.16	306.76	189.24	99,62	19.32	20.40	8.66	8.58	9.24	13.80	66,45
T19	273.24	43.15	330.49	99,6	13.59	17.06	11.32	15.86	16.64	4.60	66,71
T20	327.28	332.27	141.94	98,89	4.69	4.75	19.72	11.26	16.40	4.92	68,98

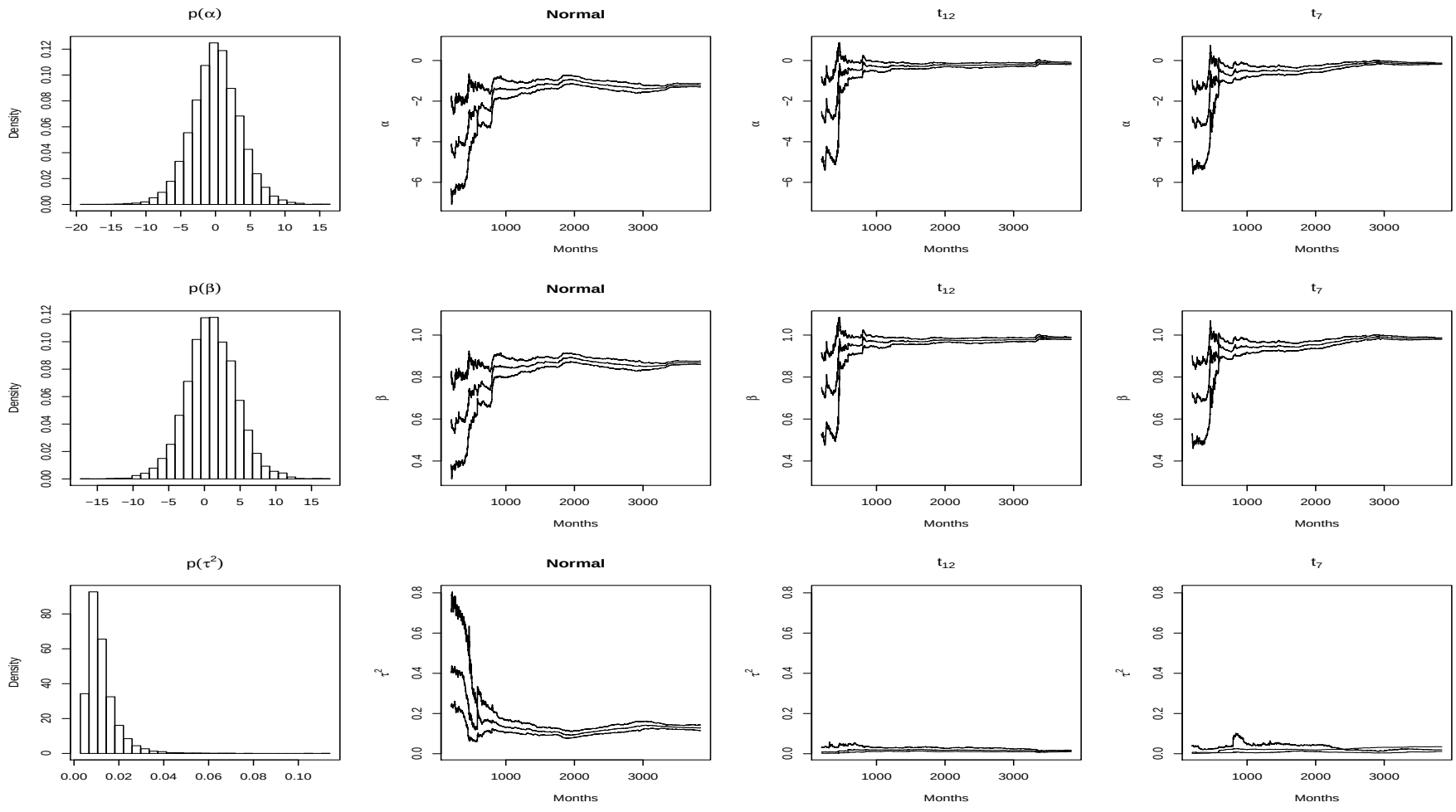
Tabela: Inefficient factor (RNI) and delay time for the SP500 return in ARSV models via PL-LW and PSLO (100 iterations and log-posterior density as the optimization function).

	LW				LW-J			PSLO			
	α	ϕ	σ	time	α	ϕ	σ	α	ϕ	σ	time
N	89.39	89.18	145.52	62.97	27.77	29.52	28.99	8.96	4.70	5.28	41.56
T2	178.89	174.00	199.76	75.91	29.55	30.60	17.50	6.50	2.20	4.20	56.80
T3	122.05	121.67	187.18	76.97	15.66	15.82	9.27	8.50	7.66	5.58	57.14
T4	151.09	149.81	240.72	77.08	33.07	37.71	16.94	11.58	4.37	5.55	57.20
T5	109.13	108.09	88.13	82.27	10.87	10.38	37.33	6.75	3.82	7.85	57.90
T6	95.92	97.71	243.37	78.92	17.62	17.46	17.33	6.99	6.63	7.82	58.17
T7	159.94	160.26	274.52	77.93	24.40	28.98	33.71	7.92	5.57	3.97	57.26
T8	95.23	95.16	262.69	77.69	23.75	23.60	25.38	3.54	4.41	3.21	56.77
T9	134.18	135.51	102.46	78.42	12.55	13.83	10.01	16.63	7.19	6.50	57.19
T10	116.39	117.29	276.67	78.23	30.86	30.94	29.29	16.63	7.19	6.50	57.19
T11	101.00	101.24	175.60	77.70	3.14	3.18	12.58	8.38	3.22	7.50	56.52
T12	100.29	97.82	100.50	77.40	25.62	25.71	25.51	13.41	4.28	3.97	56.57
T13	93.67	95.02	277.42	78.14	24.88	24.26	32.71	10.88	3.05	5.63	56.43
T14	94.17	95.08	188.17	77.40	8.36	8.66	15.84	11.50	7.35	2.70	56.48
T18	86.20	86.35	259.50	77.63	15.38	15.03	23.50	10.26	3.48	7.51	56.50
T19	102.81	103.88	302.57	78.05	34.43	35.35	33.68	11.83	3.09	6.39	56.36
T20	98.42	99.78	198.10	77.67	8.42	5.92	28.46	19.88	8.05	8.38	56.01

Tabela: Inefficient factor (RNI) and delay time for the IBOVESPA (IBV)return in ARSV models via PL-LW and PSLO (100 iterations and log-posterior density as the optimization function).

	α	ϕ	σ^2	time		α	ϕ	σ^2	time
N	0,0051	0,9895	0,0606		T11	0,0108	0,9861	0,0768	
	1,04E-02	1,30E-03	1,70E-02	46,06		2,00E-02	1,33E-02	3,91E-02	72,88
T2	0,0119	0,9891	0,0666		T12	0,0027	0,9898	0,0538	
	2,35E-02	2,40E-03	2,49E-02	67,81		5,10E-03	4,00E-04	3,30E-03	72,63
T3	0,0097	0,9887	0,0605		T13	0,0166	0,9886	0,0643	
	1,85E-02	3,80E-03	1,71E-02	66,77		3,01E-02	3,40E-03	2,87E-02	71,85
T4	0,0046	0,9876	0,0643		T14	0,0091	0,9889	0,0659	
	7,90E-03	9,60E-03	2,75E-02	68,05		1,59E-02	3,00E-03	2,50E-02	71,78
T5	0,0093	0,9896	0,0601		T15	0,01	0,9885	0,0607	
	2,03E-02	1,00E-03	1,55E-02	74,09		1,88E-02	3,50E-03	1,90E-02	71,83
T6	0,0061	0,9892	0,0598		T16	0,0066	0,988	0,0626	
	1,26E-02	2,10E-03	2,36E-02	73,07		1,35E-02	9,20E-03	2,02E-02	71,49
T7	0,008	0,9895	0,0598		T17	0,0064	0,9884	0,0627	
	1,77E-02	1,30E-03	1,79E-02	73,22		1,62E-02	4,20E-03	1,90E-02	71,86
T8	0,0092	0,9873	0,0728		T18	0,0117	0,9898	0,0565	
	1,67E-02	5,80E-03	3,37E-02	72,07		1,93E-02	5,00E-04	7,50E-03	71,83
T9	0,0104	0,9889	0,062		T19	0,0077	0,989	0,0648	
	2,36E-02	1,80E-03	1,64E-02	72,61		1,25E-02	2,40E-03	2,47E-02	71,89
T10	0,0057	0,9879	0,0687		T20	0,0062	0,9889	0,0583	
	1,14E-02	4,80E-03	3,15E-02	72,58		1,07E-02	4,90E-03	1,53E-02	72,47

Tabela: Estimated values to the parameters in SV models via PSLO. Data set:S&P500 returns. For each model, the first row: mean, the second row: standard deviation (Stdev) and time in the PSLO process (100 iterations for log-posterior density).



Particle Filter Liu West, in the SV model for daily returns of SP500.
Sample estimates for each parameter (α , ϕ and σ^2).

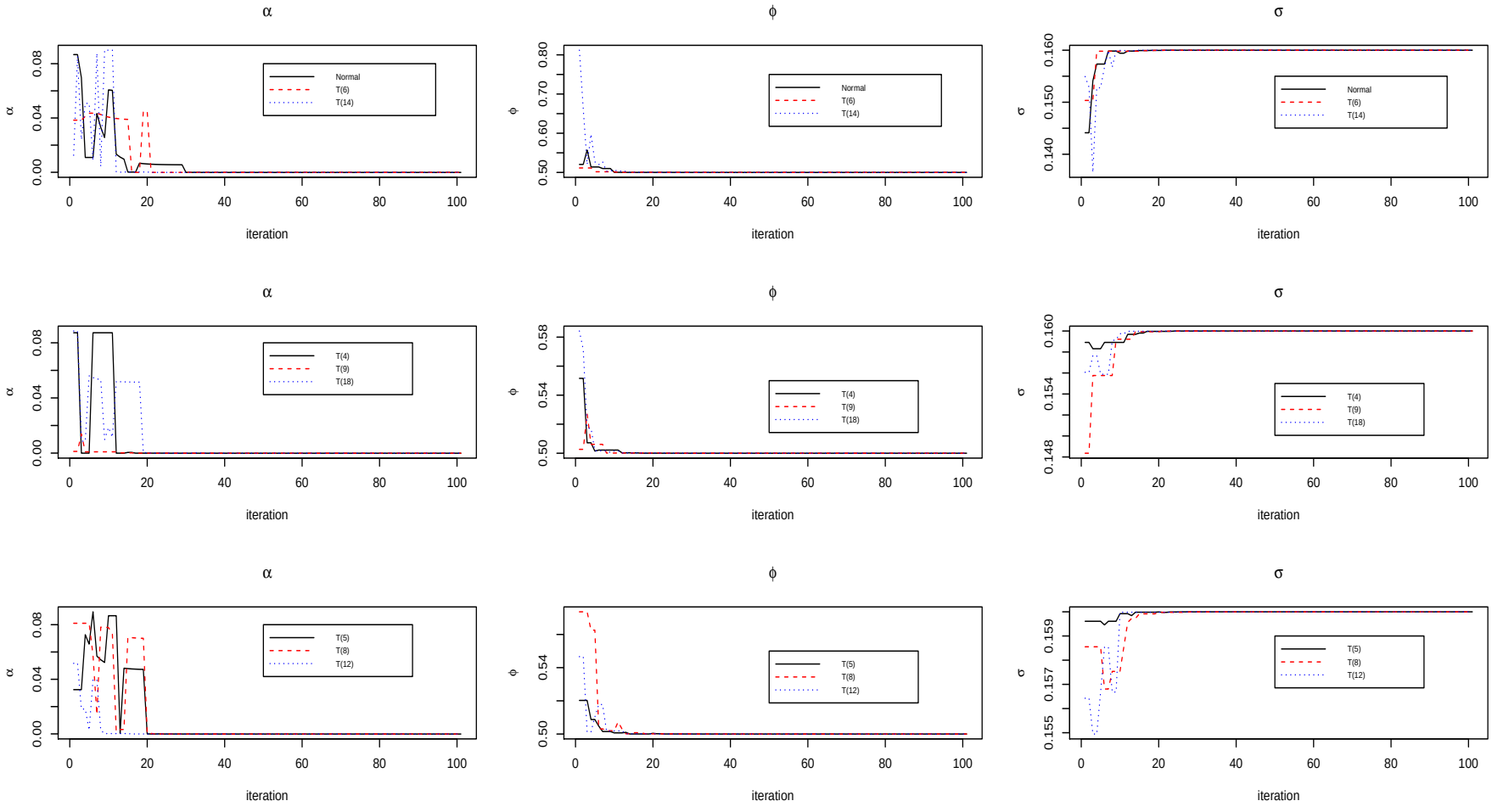
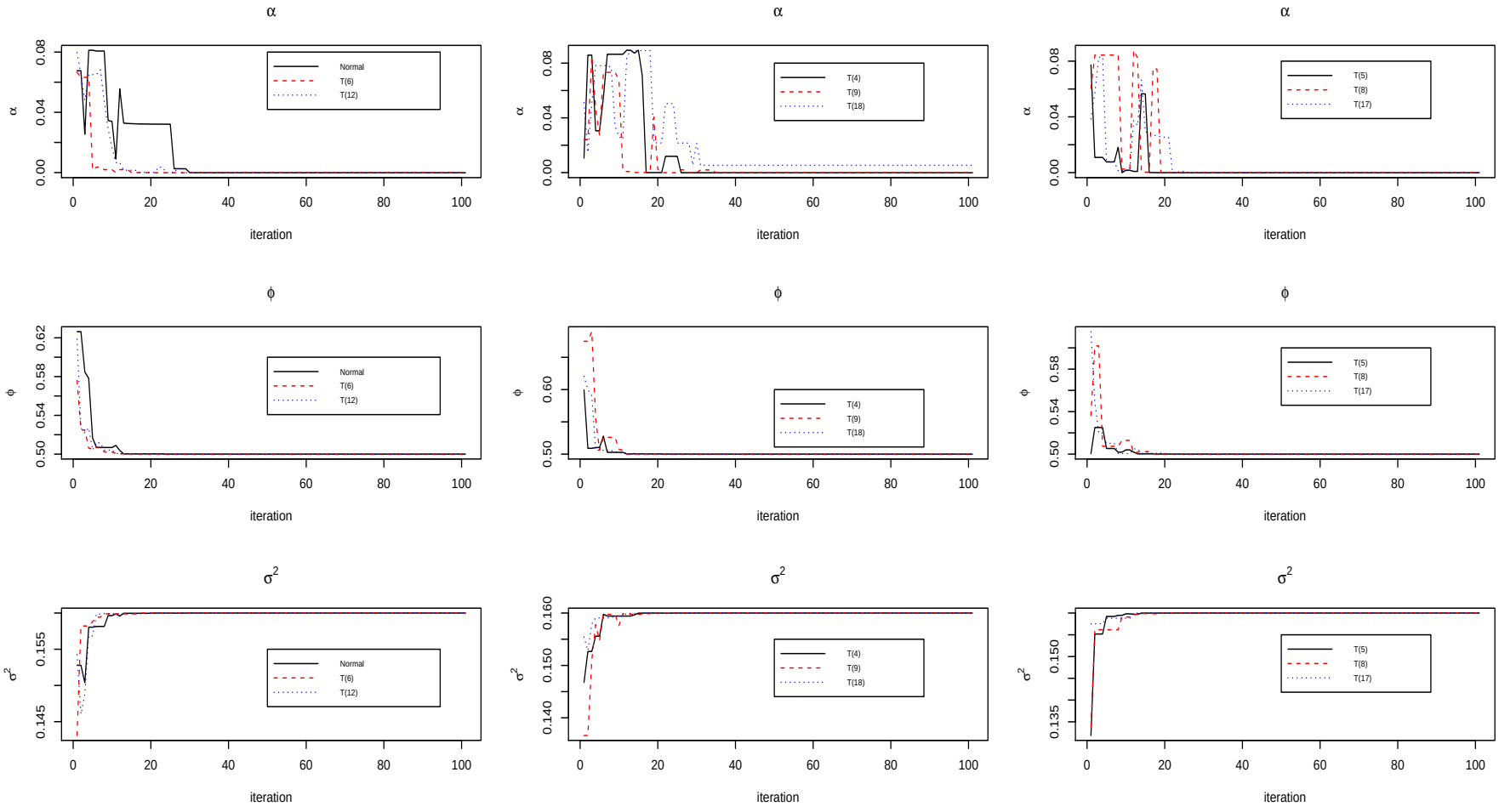


Figura: Particle Swarm learning Optimization in the SV model for daily returns of IBOVESPA. Sample estimates for each parameter (α, ϕ and σ^2) to the Normal distribution and t-distributions noise.



PSLO, in the SV model for daily returns of SP500. Sample estimates for each parameter (α , ϕ and σ^2).

Application 2: State-space AR(1) model.

A total of $T = 1000$ observations were simulated from the state-space AR(1) model with normal noise, with true parameter values $\alpha = 0.00, \phi = 0.50, \sigma^2 = 0.10$.
The hierarchical form of the model is described as,

$$
\begin{aligned}
y_t &= h_t + \epsilon_t, && \text{(13a)} \\
h_{t+1} &= \alpha + \phi h_t + \sigma_\eta \eta_t, && \text{(13b)} \\
h_0 &= N(C_0, m_0), \epsilon_t \sim N(0,1), \quad \eta_t \sim N(0,1), && \text{(13c)}
\end{aligned}
$$

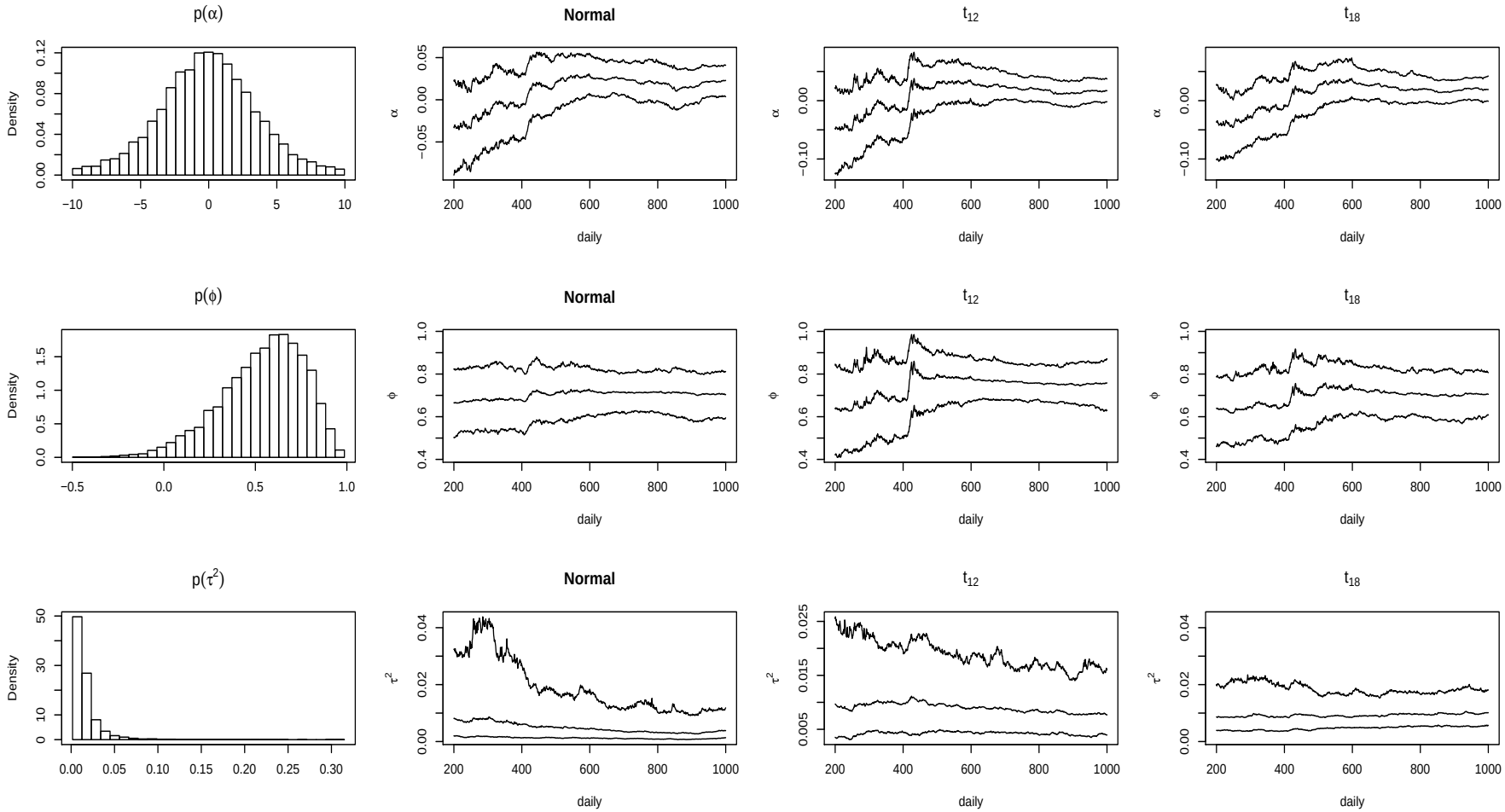
For $\epsilon_t = \lambda_t^{-\frac{1}{2}} \varepsilon_t$, ϵ_t follows a Student's-t distribution with ν degrees of freedom.

We employ the slightly informative prior on $\alpha \sim \mathcal{N}(a0 = 10, b0 = 1)$
We set $\phi = 2\phi^* - 1$, specify $\mathcal{B}eta(p, q)$ prior for ϕ^* with $p = 20$ and $q = 1.5$ which gives a prior mean for ϕ of 0.86. A prior is chosen for $\sigma^2 \sim \mathcal{IG}$ with prior mean of 0.0167 and prior standard deviation of 0.0236.
The algorithm to PSLO need the function maximized is the log-posterior density of the parameters given by,

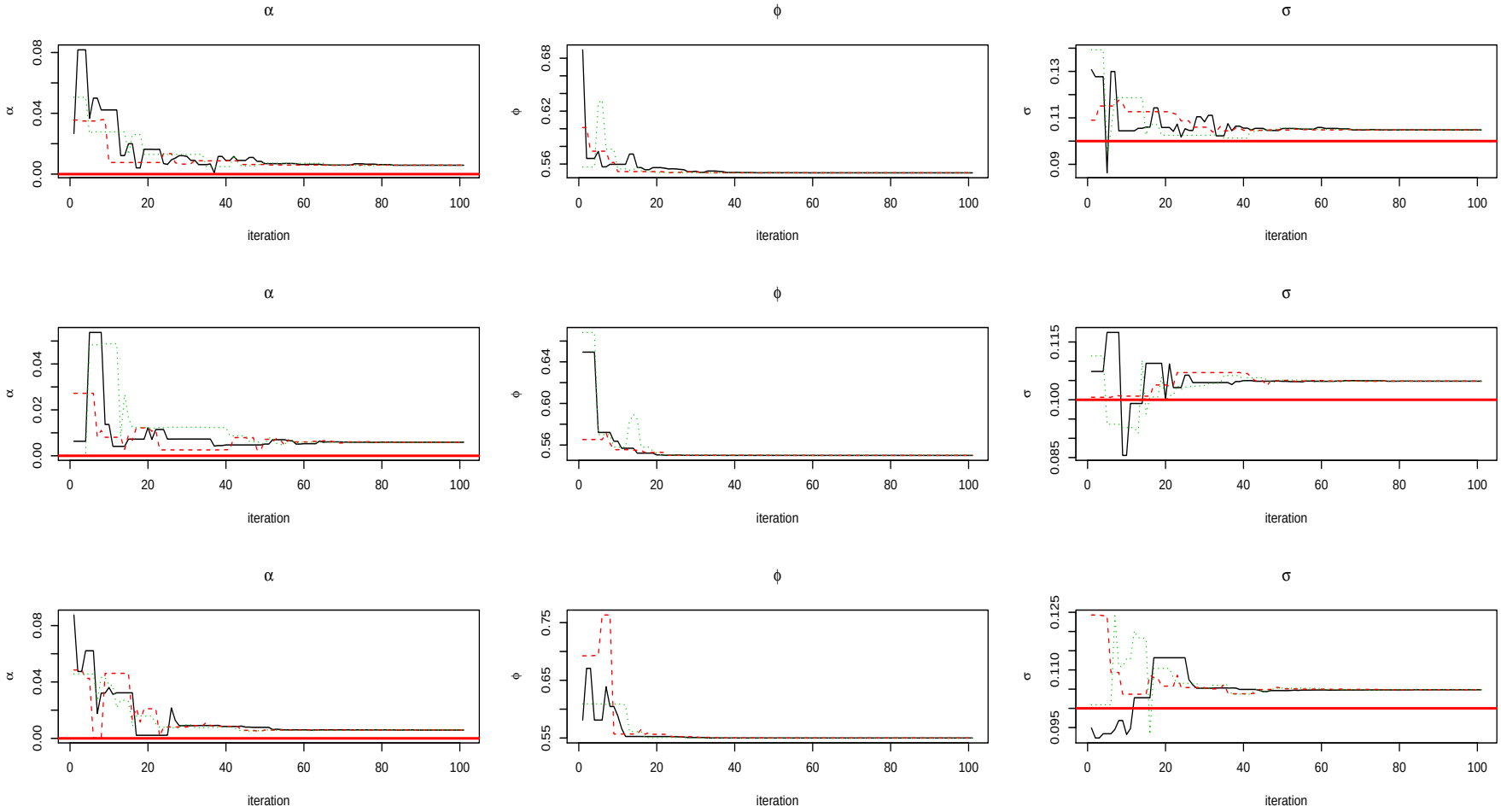
$$f(x) = \frac{(1-\phi^2)}{\sigma^2}[\frac{h_1 - \alpha}{(1-\phi)}]^2 - ln(1-\phi^2) + \sum_{t=2}^{n} \frac{(h_t - \alpha - \phi * h_{t-1})^2}{\sigma^2} + (n+To-1)ln(\sigma^2)$$

$$+(\text{Mo-2})\sigma^2 + [\frac{(\alpha - a0)}{b0}]^2 - ln(1-\phi)(2q-1) - ln(1+\phi) + 2pln(\phi)$$

To implement the sequence Monte Carlo scheme, the PL-LW was used for the simulations. We performed N=10000 particles in each time. For the PSLO methodology, we considered 30 particles in the swarm.



Particle learning used to estimate in the AR1 state-space models. Sample estimates for each parameter (α (top), β (middle) and τ^2 (botton)). The extreme lines represent the 2.5% and 97.5% quantiles and lines in the middle represent the posterior mean.



Particle Swarm learning Optimization used to estimate in the AR1 state-space models.Sample estimates for each parameter (α (top), ϕ (middle) and σ^2 (botton))

Criteria Comparison

	M-L				LPS			
	Liu West	ranking	PSLO	ranking	Liu-West	ranking	PSLO	ranking
N	9,750	17	9,308	10	12,876	4	9,290	19
T2	7,590	1	9,313	13	12,765	1	9,116	1
T3	9,546	6	9,300	2	12,859	2	9,135	2
T4	9,703	13	9,316	17	12,948	20	9,159	5
T5	9,621	9	9,302	4	12,892	9	9,150	3
T6	9,637	10	9,312	12	12,909	12	9,221	11
T7	9,396	3	9,322	18	12,916	14	9,151	4
T8	9,753	18	9,301	3	12,940	19	9,232	14
T9	9,586	8	9,325	19	12,923	16	9,198	6
T10	9,750	16	9,306	8	12,929	17	9,198	7
T11	9,559	7	9,316	16	12,890	7	9,217	10
T12	9,482	4	9,349	20	12,870	3	9,333	20
T13	9,759	20	9,316	15	12,922	15	9,223	12
T14	9,289	2	9,299	1	12,915	13	9,241	16
T15	9,710	14	9,307	9	12,901	10	9,241	15
T16	9,755	19	9,309	11	12,931	18	9,207	9
T17	9,518	5	9,304	7	12,891	8	9,247	17
T18	9,702	12	9,302	6	12,887	6	9,248	18
T19	9,675	11	9,315	14	12,905	11	9,201	8
T20	9,745	15	9,302	5	12,887	5	9,225	13

Tabela: Criteria Comparison for the Particle Learning methods (PSLO and PL-LW) with normal and Student's-t states. Time series: S&P500 returns. (Criteria: Estimated marginal likelihoods scale (log-ML) and log predictive score LPS).

Tabela: Criteria Comparison for the Particle Learning methods (PSLO and PL-LW). Time series: Ibovespa returns. (Criteria: Estimated marginal likelihoods scale (log-ML) and log predictive score LPS).

	M-L				LPS			
	Liu West	ranking	PSLO	ranking	Liu-West	ranking	PSLO	ranking
N	9,77	7	9,309	4	12,87	6	9,19	9
T2	10,54	9	9,303	2	12,86	5	9,04	2
T3	8,83	3	9,313	5	12,81	4	9,05	3
T4	9,14	5	9,302	1	12,88	7	9,01	1
T5	9,53	6	9,327	8	12,77	3	9,09	5
T6	7,46	1	9,316	6	12,69	2	9,11	6
T7	8,94	4	9,361	9	12,57	1	9,12	7
T8	7,88	2	9,308	3	12,91	8	9,07	4
T9	10,06	8	9,323	7	12,91	9	9,12	8

Conclusions

- This paper proposed an alternative method to state-space models in dynamic models, specifically in NLNG.
- The PSLO method smooths and filters the states, and estimates the parameters of the model in an efficient and accurate way.
- We understand that our contribution is the introduction of a new process that changes the Kernel process or sampler process (sequences Monte Carlo) by the optimization process.
- In addition, the major difference from the other methods is the individual and global learning, given by the swarm particles.
- The learning of each particulate of the swarm (collective and individual) makes the method faster and efficient.
- Therefore, the result is achieved by information spreading through the swarm instead of individual learning, thus producing a rapid and efficient algorithm.
- The empirical results show that the estimation of the parameters and states of the models are more accurate and more efficient in terms of computational time via PSLO rather than the PF.

Gracias.

Bibliography

GORDON, N. J.; SALMOND, D. J.; SMITH, A. F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. v. 140, n. 2, p. 107–113, 1993.

LIU, J.; WEST, M. Combined parameter and state estimation in simulation-based filtering. In: *Sequential Monte Carlo methods in practice*. [S.l.]: Springer, 2001. p. 197–223.

SHEPHARD, N.; PITT, M. Likelihood analysis of non-Gaussian measurements time series. *Biometrika*, v. 84, p. 653–667, 1997.

SORENSON, H. W.; ALSPACH, D. L. Recursive bayesian estimation using gaussian sums. *Automatica*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 7, n. 4, p. 465–479, 1971. ISSN 0005-1098.

TAYLOR, S. J. *Financial returns modelled by the product of two stochastic processes a study of the daily sugar prices, 1961-79*. In Time Series Analysis: Theory and Practice: (ed. O. D. Anderson). New York: Elsevier Science Publishing Co., Amsterdam: North-Holland Publishing Co., 1982. v. 1.

WEST, M.; HARRISON, J. *Bayesian forecasting and dynamic models*. Springer-Verlag: New York., 1997.