# TRELLIS-BASED SEARCH OF THE MAXIMUM A POSTERIORI SEQUENCE USING PARTICLE FILTERING

*Tanya Bertozzi* [*†], *Didier Le Ruyet*[†], *Gilles Rigal* [*] *and Han Vu-Thien* [†]

[*] DIGINEXT, 45 Impasse de la Draille, 13857 Aix en Provence Cedex 3, France, Email: bertozzi@diginext.fr

[†] CNAM, 292 rue Saint Martin, 75141 Paris Cedex 3, France

## ABSTRACT

For a given computational complexity, the Viterbi algorithm applied on the discrete representation of the state space provided by a standard particle filtering, outperforms the particle filtering. However, the computational complexity of the Viterbi algorithm is still high. In this paper, we propose to use the M and T algorithms in order to reduce the computational complexity of the Viterbi algorithm and we show that these algorithms enable a reduction of the number of particles up to 20%, practically without loss of performance with respect to the Viterbi algorithm.

## 1. INTRODUCTION

Many real systems of data analysis require the estimation of unknown quantities from measures provided by sensors. In general, the physical phenomenon can be represented by a mathematical model, which describes the time evolution of the unknown quantities called hidden state and their interactions with the observations. Often, the observations arrive sequentially in time and it is of interest to update at each instant the estimation of the hidden state. Except in a few special cases, including linear Gaussian state space models and hidden finite-state space Markov chains, it is impossible to derive an exact analytical solution to the problem of sequential estimation of the hidden state. For over thirty years, many approximation schemes have been proposed to solve this problem and recently, the approach which receives the major interest is based on the particle filtering techniques [1]. These methods allow to approximate iteratively the posterior distribution of the hidden state given the observations by weighted points or particles which evolve in the state space. Therefore, the particle filtering gives a discrete approximation of the state space of a continuous state space model.

In [2], the estimation of the hidden state using a standard particle filtering is compared to the estimation done by the Viterbi Algorithm (VA) [3]-[4], where the trellis is built from the discrete representation of the state space provided by the particle filtering. For a given computational complexity, the VA outperforms the standard particle filtering. However, the computational complexity of this solution is still high since the VA analyzes all the possible paths arriving to each particle.

In this paper, we propose to apply the M algorithm [5] and the T algorithm [6] in order to reduce the computational complexity of the VA built on the particle states. This paper is organized as follows. In Section II the system model is presented. The structure of the standard particle filtering is introduced in Section III. Section IV describes the VA, the M and the T algorithms built on the particle states. Finally, simulation results are given in Section IV.

## 2. THE STATE SPACE MODEL

The standard Markovian state space model is represented by the following expressions:

$$\begin{cases} x_k &= f(x_{k-1}, w_k) \\ y_k &= h(x_k, v_k) \end{cases}, \tag{1}$$

where $k \geq 1$ is a discrete time index, $w_k$ and $v_k$ are independent white noises. The functions $f$ and $h$ can involve nonlinearity and the noises $w_k$ and $v_k$ can be non Gaussian. The first equation describes the time evolution of the hidden state $x_k$ and the second equation shows the interactions between the observation $y_k$ and the hidden state. In this paper, we consider the filtering problem yielding the estimation of the hidden state $x_t$ at a time $t$ from the observations $y_{1:t} = \{y_1, \cdots, y_t\}$. The estimation of the hidden state can be obtained by the Minimum Mean Square Error (MMSE) method or by the Maximum A Posteriori (MAP) method. The MMSE solution is given by the following expectation:

$$\hat{x}_t = E[x_t|y_{1:t}]. \tag{2}$$

The calculation of (2) involves the knowledge of the filtering distribution $p(x_t|y_{1:t})$. When this distribution is multimodal, the MMSE estimate is located between the modes and is far from the true value of the hidden state. In this case, it is preferable to use the MAP method, which provides the estimate of the hidden state sequence $x_{1:t} = \{x_1, \cdots, x_t\}$:

$$\hat{x}_{1:t} = \arg\max_{x_{1:t}} p(x_{1:t}|y_{1:t}). \tag{3}$$

The calculation of (3) requires the knowledge of the posterior distribution $p(x_{1:t}|y_{1:t})$.

## 3. THE STANDARD PARTICLE FILTERING

The aim of the standard particle filtering is to approximate recursively in time the posterior distribution $p(x_{1:t}|y_{1:t})$ with weighted particles:

$$p(x_{1:t}|y_{1:t}) \approx \sum_{i=1}^{N} \tilde{w}_t^i \delta(x_t - x_t^i) \cdots \delta(x_1 - x_1^i), \tag{4}$$

where N is the number of particles, $\tilde{w}_t^i$ is the normalized weight associated with the particle $i$ and $\delta(x_k - x_k^i)$ denotes the Dirac delta centered in $x_k = x_k^i$ for $k = 1, \cdots, t$. The iteration is achieved by evolving the particles from time 1 to time $t$, using the Sequential Importance Sampling and Resampling (SISR) methods [7]. In

general, an initial distribution $p(x_0)$ of the hidden state is available. Initially, the supports $\{x_0^i; i = 1, \cdots, N\}$ of the particles are drawn according to the initial distribution. The evolution of the particles from time $k$ to time $k+1$ is achieved with an importance sampling distribution [8]. At each time $k$ the particles are drawn according to the importance function $\pi(x_k|x_{0:k-1}, y_{1:k})$. The importance function enables to calculate recursively in time the weights associated with the particles:

$$w_k^i = w_{k-1}^i \frac{p(y_k|x_k^i)p(x_k^i|x_{k-1}^i)}{\pi(x_k^i|x_{0:k-1}^i, y_{1:k})}, \quad (5)$$

where $k \geq 1$, $i = 1, \cdots, N$ and $w_0^i = 1/N, \forall i$. the normalized weights are given by:

$$\tilde{w}_k^i = \frac{w_k^i}{\sum_{j=1}^N w_k^j}. \quad (6)$$

This algorithm presents a degeneracy phenomenon. After a few iterations of the algorithm, only a particle has a normalized weight almost equal to 1 and the other weights are very close to zero. This problem of the SIS method can be eliminated with a resampling of the particles. A measure of the degeneracy is the effective sample size $N_{eff}$ [9]-[10], estimated by:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (\tilde{w}_k^i)^2}. \quad (7)$$

When $\hat{N}_{eff}$ is below a fixed threshold $N_{thres}$, the particles are resampled according the weight distribution [7]. After each resampling task, the normalized weights are initialized to $1/N$.

The optimal importance function, which minimizes the degeneracy of the SIS algorithm, is given by:

$$\pi(x_k|x_{0:k-1}, y_{1:k}) = p(x_k|x_{k-1}, y_k). \quad (8)$$

In the general case of nonlinear non Gaussian state space model, (8) cannot be evaluated in an analytical form. It is only possible to calculate (8) exactly, when the noises $w_k$ and $v_k$ are Gaussian and the function $h$ is linear. If $w_k$ and $v_k$ are Gaussian and $h$ is nonlinear, we can obtain an approximation of (8) by linearizing the function $h$ in $x_k = f(x_{k-1}, w_k)$ [7]. A simpler choice for the importance function is represented by the prior distribution:

$$\pi(x_k|x_{0:k-1}, y_{1:k}) = p(x_k|x_{k-1}), \quad (9)$$

however, this method can be inefficient since the state space is explored a priori without taking account of the observations.

Using the SISR methods, we can provide a MMSE of the hidden state at each time $k$:

$$\begin{aligned} \hat{x}_k &= \int x_k p(x_k|y_{1:k}) dx_k \\ &= \int x_k \sum_{i=1}^N \tilde{w}_k^i \delta(x_k - x_k^i) dx_k \\ &= \sum_{i=1}^N x_k^i \tilde{w}_k^i. \end{aligned} \quad (10)$$

For the MAP estimate, the maximization in (3) is only performed on the $N$ sequences of particles. Applying the Bayes theorem to the posterior distribution at a time $k$:

$$p(x_{1:k}|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|x_{k-1})}{p(y_k|y_{1:k-1})} p(x_{1:k-1}|y_{1:k-1}), \quad (11)$$

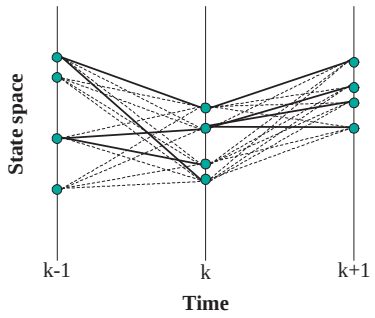


**Fig. 1**. Application of the VA in a particle trellis ($N = 4$).

we observe that the posterior distribution in (3) associated at each particle can be processed iteratively:

$$\lambda_k^i = \lambda_{k-1}^i + \ln p(y_k|x_k^i) + \ln p(x_k^i|x_{k-1}^i), \quad (12)$$

where we have omitted the normalization term identical for each particle and $\lambda_k^i$ denotes the metric of the particle $i$ at time $k$. At time $t$, the MAP estimate coincides with the path in the state space of the particle with maximum $\lambda_t^i$.

## 4. COMPLEXITY REDUCTION OF THE VITERBI ALGORITHM

The VA, introduced by Viterbi in 1967 [3] and analyzed in detail by Forney in 1973 [4], is a dynamic programming algorithm, which provides an iterative way of finding the most probable sequence in the MAP sense of hidden states of a finite-state discrete-time Markov model. It reduces the complexity of the problem by avoiding the necessity to examine every path through the trellis. However, in the most general case of a continuous-state space model, the VA cannot be applied. In [2], the authors have proposed to perform the VA on the discrete trellis built by a SISR technique. Each particle represents a state with a metric expressed by (12). An example of a particle trellis is represented in Fig. 1. We consider the generic transition from time $k-1$ to time $k$. At time $k$, the VA analyzes all the possible paths which reach the arrival particle $p_a$, for $p_a = 1, \cdots, N$. The metric associated with a possible path in the particle trellis from a departure particle $p_d$ at time $k-1$ to $p_a$ is given by:

$$\lambda_k^{p_a} = \lambda_{k-1}^{p_d} + \ln p(y_k|x_k^{p_a}) + \ln p(x_k^{p_a}|x_{k-1}^{p_d}). \quad (13)$$

Among these paths from all the $p_d$ to $p_a$, only the path with the maximum metric is kept. At the final instant $t$, the MAP estimate of the hidden state sequence coincides with the path of the particle with maximum metric. If the computational complexity of the SISR algorithm is proportional to the number $N$ of particles, the computational complexity of the VA is proportional to $N^2$. In [2], the authors have shown that the VA processed on a trellis of $N$ particles outperforms a SISR algorithm with $N^2$ particles. The problem is that $N^2$ can assume very high values. In this paper, we propose to reduce the computational complexity of the VA using the M and T algorithms, while keeping the same performance.

The M algorithm retains the $M$ best paths, with $M$ less than the total number of states, from one iteration to the next one. In the other hand, the T algorithm keeps variable number of paths

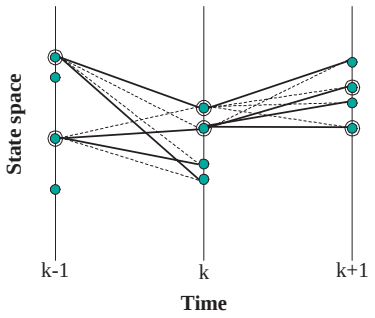


**Fig. 2**. Application of the M algorithm in a particle trellis ($N = 4$, $M = 2$).

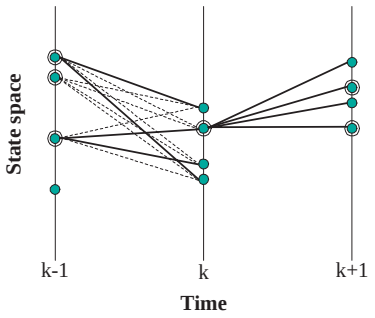


**Fig. 3**. Application of the T algorithm in a particle trellis ($N = 4$, $\overline{N} = 2$).

depending on the threshold parameter $T$. First, let's modify the M algorithm on the particle trellis built by the SISR algorithm.

At time 1, we consider all the possible paths from the departure particles $p_d$ to the arrival particles $p_a$ and we retain one path for each $p_a$, as in the VA. At that time, we introduce a new step. From the $N$ arrival particles we keep the $M$ particles with the best metrics, where $M < N$. At the next time 2, the number of departure particles is $M$ and of arrival particles is $N$. Therefore, only $MN$ paths from $p_d$ to $p_a$ are possible. At time 2, we retain the $M$ particles with the best metrics and go through the trellis in this way up to the final time $t$. The path of the particle with maximum metric at time $t$ represents the MAP estimate of the hidden state sequence, as in the VA. This M algorithm has a computational complexity proportional to $MN$. An example is shown in Fig. 2.

Let's consider now the T algorithm. At time 1, we perform the VA. Then, among the arrival particles we determine the particle with the maximum metric. We calculate the difference between the maximum metric and metrics of the other arrival particles. When this difference is greater than a given threshold $T$, the particle is discarded. At the next time 2, the departure particles are $N_1 <= N$ and the arrival particles are $N$. As the consequence, only consider $N_1 N$ paths from $p_d$ to $p_a$. At time 2, we retain the $N_2$ particles which have survived the threshold test and go through the trellis in this way up to the final time $t$. The path of the particle with maximum metric at time $t$ represents the MAP estimate of the hidden state sequence, as in the VA. This T algorithm has a mean computational complexity proportional to $\overline{N}N$, where $\overline{N}$ is the mean number of survivor particles at each instant. An example is given in Fig. 3.

## 5. SIMULATION RESULTS

In order to compare the simulation results of the standard SISR, Viterbi, M and T algorithms, we consider the following nonlinear Gaussian state space model [11]-[12]-[2]:

$$\left\{ \begin{array}{rcl} x_k & = & \frac{1}{2}x_{k-1} + 25\frac{x_{k-1}}{1+x_{k-1}^2} + 8\cos(1.2k) + w_k \\ y_k & = & \frac{x^2}{20} + v_k \end{array} \right. , \quad (14)$$

where the time index $1 \leq k \leq t$ with $t = 200$, the density of the initial hidden state $x_0$ is Gaussian with zero mean and variance 5 and $w_k$ and $v_k$ are mutually independent white Gaussian noises with zero mean and variance respectively equal to 10 and

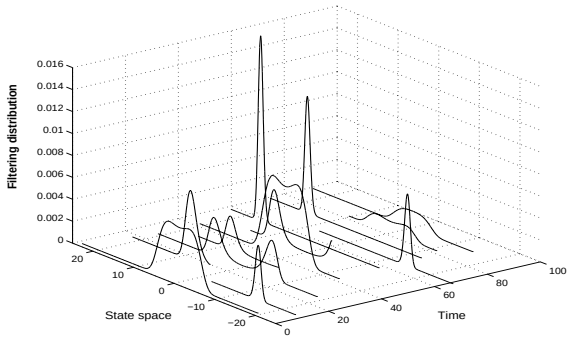


**Fig. 4**. Filtering distribution.

1. In this case, the filtering distribution $p(x_k|y_{1:k})$ can be bimodal, as illustrated in Fig. 4. This figure is obtained by applying a SISR with $N = 1000$ particles, an importance function $\pi(x_k|x_{0:k-1}, y_{1:k}) = p(x_k|x_{k-1})$ and a resampling step made at each time (bootstrap filter, [11]).

To evaluate the performance of the different algorithms exposed above, we use the mean of the absolute value of the filtering error on $N_r = 100$ realizations of the algorithms:

$$m_{err} = \frac{1}{N_r}\sum_{n=1}^{N_r} m_{err}(n) = \frac{1}{N_r}\sum_{n=1}^{N_r}\frac{1}{t}\sum_{k=1}^{t}|x_k(n) - \hat{x}_k(n)|, \quad (15)$$

where $m_{err}(n)$ is the mean of the absolute value of the error filtering for the realization $n$ and $x_k(n)$ is the hidden state at time $k$ for the realization $n$, and the variance of this error:

$$\begin{array}{rcl} \sigma_{err}^2 & = & \frac{1}{N_r}\sum_{n=1}^{N_r}\sigma_{err}^2(n) \\ & = & \frac{1}{N_r}\sum_{n=1}^{N_r}\frac{1}{t}\sum_{k=1}^{t}\Big(|x_k(n)-\hat{x}_k(n)|-m_{err}(n)\Big)^2 \end{array} \quad (16)$$

where $\sigma_{err}^2(n)$ is the variance of the absolute value of the error filtering for the realization $n$. For the standard particle filtering, we use a SISR algorithm with an importance function calculated by linearizing the observation model (14) and a resampling threshold $N_{thres} = N/5$. We consider only the MAP estimate which in this case gives the better performance because the filtering distribution

| | SISR MAP algorithm | | Viterbi algorithm | |
|---|---|---|---|---|
| $N$ | Mean | Variance | Mean | Variance |
| 100 | 0.906583 | 3.029062 | 0.870553 | 5.982229 |
| 250 | xxxx | xxxx | 0.849045 | 5.219399 |
| 500 | xxxx | xxxx | 0.804020 | 3.936105 |
| 1000 | xxxx | xxxx | 0.784171 | 3.394146 |

**Table 1**. Performance of the SISR and Viterbi algorithms.

| | M algorithm | | |
|---|---|---|---|
| $N$ | $M$ | Mean | Variance |
| 100 | 90 | 0.878995 | 6.145979 |
| | 80 | 0.910603 | 6.684327 |
| 250 | 225 | 0.853668 | 5.273226 |
| | 200 | 0.871307 | 5.648677 |
| 500 | 450 | 0.808492 | 4.013840 |
| | 400 | 0.834472 | 4.514141 |
| 1000 | 900 | 0.795176 | 3.545520 |
| | 800 | 0.807538 | 3.827843 |

**Table 2**. Performance of the M algorithm.

can be bimodal. The obtained results are shown in Tables 1, 2 and 3. In Table 1, the SISR algorithm and the VA have the same computational complexity proportional to $N^2$, since the first one uses $N^2$ particles. We notice that for a given computational complexity, the VA outperforms the standard particle filtering. In Table 2, the computational complexity of the M algorithm is proportional to $MN$. If $M$ is less than $N$ by 10%, we have almost the same performance than the VA. Nevertheless, if we reduce the number of particles by 20%, the performance degrades. In Table 3, the mean computational complexity of the T algorithm is proportional to $\overline{N}N$. We observe that the T algorithm presents better performance than the one of the M algorithm. Applying the T algorithm, we can reduce the number of particles up to nearly 20% practically without loss of performance with regard to the VA.

## 6. CONCLUSION

In this paper, we have analyzed the problem of the estimation of a nonlinear non Gaussian hidden state, solved generally with the SISR algorithm. The particles of the SISR algorithm provide a discrete representation of the state space. If we see the particles as the states of a trellis, we can search for the most likely sequence using the VA. We have shown that for a given complexity, the VA outperforms the SISR algorithm. However, the computational complexity of this solution is still high. As the consequence, we have proposed the M and the T algorithms in order to reduce the computational complexity of the VA. We can conclude that these algorithms enable a reduction of the number of particles up to 20%, practically without loss of performance with regard to the VA.

| | T algorithm | | |
|---|---|---|---|
| $N$ | $\overline{N}$ | Mean | Variance |
| 100 | 86.461960 | 0.873065 | 6.048468 |
| | 77.598141 | 0.886633 | 6.445502 |
| 250 | 217.145276 | 0.850905 | 5.271618 |
| | 195.566131 | 0.861759 | 5.637627 |
| 500 | 434.812261 | 0.806332 | 4.020476 |
| | 391.834975 | 0.824472 | 4.598820 |
| 1000 | 869.416231 | 0.786935 | 3.480737 |
| | 784.061206 | 0.810000 | 4.154820 |

**Table 3**. Performance of the T algorithm.

## 7. REFERENCES


[1] A. Doucet, J. F. G. de Freitas and N. J. Gordon, *Sequential Monte Carlo methods in practice*. New York: Springer-Verlag, 2001.

[2] S. Godsill, A. Doucet and M. West, "Maximum a posteriori sequence estimation using Monte Carlo particle filters," *Annals of the Institute of Statistical Mathematics*, Vol. 52, No. 1, pp. 82–96, 2001.

[3] A. Viterbi, "Error bounds for convolutional coding and an asymptotically optimum decoding algorithm," *IEEE Trans. on Inform. Theory*, Vol. 13, pp. 260–269, Apr. 1967.

[4] G. D. Forney, "The Viterbi algorithm," *Proc. of the IEEE*, Vol. 61, No. 3, March 1973.

[5] J. B. Anderson and S. Mohan, "Sequential coding algorithm: a survey and cost analysis," *IEEE Trans. on Com.*, Vol. 32, pp. 169–176, Feb. 1984.

[6] S. J. Simmons, "Breadth-first trellis decoding with adaptive effort," *IEEE Trans. on Com.*, Vol. 38, pp. 3–12, Jan. 1990.

[7] A. Doucet, S. Godsill and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, Vol. 10, No. 3, pp. 197–208, 2000.

[8] J. Geweke, "Bayesian inference in econometric models using Monte Carlo integration," *Econometrica*, Vol. 24, pp. 1317–1399, 1989.

[9] A. Kong, J. S. Liu and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *Journ. of the American Statistical Association*, Vol. 89, pp. 278–288, 1994.

[10] J. S. Liu, "Metropolized independent sampling with comparison to rejection sampling and importance sampling," *Statistics and Computing*, Vol. 6, pp. 113–119, 1996.

[11] N. J. Gordon, D. J. Salmond and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc.*, Vol. 140(2), pp. 107–113, 1993.

[12] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journ. of Computational and Graphical Statistics*, Vol. 5(1), pp. 1–25, 1996.