Deterministic Resampling
for Sequential Monte Carlo Methods

Anonymous Author(s)
Affiliation
Address
email

Abstract

Sequential Monte Carlo methods are a general class of sampling-based methods, which rely on a combination of sequential importance sampling and resampling. Although powerful resampling schemes exist, they often remove so many samples. This causes increase of internal correlations and loss of diversity among samples. To overcome this drawback, we introduce deterministic resampling in this paper. This scheme neither employs any randomness nor resets weights after resampling. It adopts a completely deterministic approach to resampling and keep the weights. We show the proposed scheme never removes more than half samples in any case. This greatly improves approximation performance as our simulation results show.

1 Introduction

Sequential Monte Carlo (SMC) methods are a general class of sampling-based methods designed to achieve computational efficiency in dynamic problems which include intractable computations. It is achieved by the statistical online approximation of the intractable computations using weighted random samples drawn from a probability density of interest. These methods were originally invented in the early 1950s by physicists [1] and have become very popular with engineering scientists since the introduction of resampling [2]. Especially as the name of Particle Filter, these methods are now popularly used to solve recursive Bayesian estimation problems repeatedly arising in many engineering areas such as signal processing, AI, control, machine learning, computer vision, or robotics.

The basis of SMC methods is Sequential Importance Sampling (SIS) developed from Monte Carlo (MC) techniques [3]. Although it provides the main key to success of SMC methods, SIS always fails to provide good approximation performance because many samples become degenerated after some SIS iterations [4]. To ease this, SMC methods generally require resampling [5]. Resampling eases this by removing highly degenerated samples [6], and there now exist powerful resampling schemes such as systematic resampling [7] and residual resampling [8]. These schemes, however, still have one critical drawback: They often remove so many samples. This causes many serious practical and theoretical problems such as the increase of internal correlations and the loss of diversity [5, 8–15].

In this paper, we introduce another resampling scheme, which we call as *Deterministic Resampling*, to overcome the drawback. A main difference between this scheme and the others is that this scheme never employs any randomness, rather, it adopts a completely deterministic approach to resampling for managing the drawback. One more difference is that it never resets weights but keep the weights properly after resampling. Our analysis shows that the number of removed samples never exceeds half of the sample size in any case, when the proposed scheme is used. Consequently, significant improvements in sampling efficiency and approximation performance can be obtained compared with current resampling schemes. We show these results through two simulation studies. Note that our proposed scheme can be used in any advanced SMC methods just like other resampling schemes.

The rest of this paper is structured as follows. In Section 2, we review SMC methods in brief. Then, the basics and method of deterministic resampling are presented with the algorithm in Section 3. In Section 4, we evaluate our scheme using two simulation models. In Section 5, we draw a conclusion.

2 Sequential Monte Carlo Methods

The combination of SIS and resampling has been widely regarded as standard SMC methods in the most of literature and many advanced versions have been proposed based on the standard methods for some improvements [1, 5, 12, 16]. This section reviews the standard SMC methods.

2.1 Sequential Importance Sampling

Suppose we are interested in computing an expectation $E_p(f) = \int f(x)p(x)\,\mathrm{d}x$ where $f(x)$ is a function of interest and $p(x)$ is a target density. If we draw N i.i.d. samples $\{x^i\}_{i=1}^N$ from the target density[1], these samples can be used to approximate the target density as $p_N(x) = \frac{1}{N}\sum_{i=1}^N \delta_{x^i}(x)$ where $\delta_{x^i}(x)$ denotes the Dirac delta mass located at x^i. Then we can approximate the expectation as $E_{p_N}(f) = \frac{1}{N}\sum_{i=1}^N f(x^i)$ which is obviously unbiased and approaches $E_p(f)$ as $N \to \infty$.

Now, consider a slightly modified approach. We propose a density $q(x)$ and draw N i.i.d. samples $\{x^i\}_{i=1}^N$ from the proposed density instead the target density. If we assign a weight w^i to a sample x^i for all i such that the set of N particles $\{x^i, w^i\}_{i=1}^N$ is proper with respect to $p(x)$ [8], the expectation can be estimated by $E_{q_N}(f) = \sum_{i=1}^N w^i f(x^i)$ which also approaches $E_p(f)$ as $N \to \infty$.

Although the first approach is perfect for MC simulation, the modified approach can be beneficial to many real problems. For example, if we set $w^i = w(x^i)/\sum_{j=1}^N w(x^j)$ where $w(x) \propto p(x)/q(x)$, then MC methods can be applied to a problem whose target density is unusable to draw samples but only known up to a normalizing constant. Furthermore, if we select $q(x)$ to be biased toward important regions where the contribution to approximation is relatively high compared to other regions, we can obtain high sampling efficiency (i.e., super-efficient) [1, 3]. Such a case of the modified approach is known as Importance Sampling (IS) with the names of $q(x)$: importance density, w^i: normalized importance weight, and $w(x^i)$: unnormalized importance weight.

Suppose we are now interested in computing an expectation sequentially in a dynamic system which evolves by a sequence of target densities $\{p_t(x_{0:t})\}$ with increasing dimension $x_{0:t} = (x_{0:t-1}, x_t)$ for all discrete time $t = 1, 2, \ldots$ as follows.

$$E_{p_t}(f_t) = \int f_t(x_{0:t})p_t(x_{0:t})\,\mathrm{d}x_{0:t} \tag{1}$$

where $f_t(x_{0:t})$ is a function of interest at time t. We can approximate the expectation at time t using a sequential version of IS which is known as Sequential Importance Sampling (SIS). The mechanisms between SIS and IS are exactly same except that we now have decompositions

$$p_t(x_{0:t}) = p_{t-1}(x_{0:t-1})p_t(x_t|x_{0:t-1}) \qquad \text{and} \qquad q_t(x_{0:t}) = q_{t-1}(x_{0:t-1})q_t(x_t|x_{0:t-1}) \tag{2}$$

and have the set of N particles $\{x_{0:t}^i, w_t^i\}_{i=1}^N$ where the normalized importance weight is given by

$$w_t^i = \frac{w_t(x_{0:t}^i)}{\sum_{j=1}^N w_t(x_{0:t}^j)} \quad \text{where} \quad w_t(x_{0:t}) \propto \frac{p_t(x_{0:t})}{q_t(x_{0:t})} \propto w_{t-1}(x_{0:t-1})\frac{p_t(x_t|x_{0:t-1})}{q_t(x_t|x_{0:t-1})} \tag{3}$$

Then the expectation at time t can be estimated by

$$E_{(q_N)_t}(f_t) = \sum_{i=1}^N w_t^i f_t(x_{0:t}^i) \tag{4}$$

which approaches (1) as $N \to \infty$.

However, it has been shown [4] that the variance of weights in SIS can only increase over time, which indicates that almost samples will have negligible weights after a few time iterations. It implies that only few samples contribute to approximation and a large computational effort is wasted on samples whose contributions to approximation are almost zero, which results in poor approximation and low sampling efficiency. To alleviate this undesirable phenomenon of SIS which is called as a sample degeneracy problem throughout this paper, resampling is an essential step in SMC methods.

[1] For simplicity, we shall allow ourselves to use slight abuses of random varaiable and density notations.

2.2 Resampling

Resampling for SMC methods was originally introduced in 1993 [2], motivated by the Bootstrap methods [17]. The basic idea of resampling is to remove samples that have negligible weights. Then we can save the computational effort which would be wasted by those samples, and devote it to samples whose contributions to approximation are relatively large. Obviously, this alleviates the sample degeneracy problem and improves sampling efficiency and approximation performance.

Suppose we have a set of N particles $\{x_{0:t}^i, w_t^i\}_{i=1}^N$ at time t after SIS. Then resampling proceeds as

Step 1. Select N samples $\{\hat{x}_{0:t}^j\}_{j=1}^N$ with replacement from $\{x_{0:t}^i\}_{i=1}^N$ according to $\{w_t^i\}_{i=1}^N$
Step 2. Assign a proper weight $\hat{w}_t^j$ to a sample $\hat{x}_{0:t}^j$ for all j

Now, we have a set of N particles $\{\hat{x}_{0:t}^j, \hat{w}_t^j\}_{j=1}^N$ after resampling. Since the number of times that a sample $x_{0:t}^i$ is selected depends upon its associated weight w_t^i, it is clear that this procedure gives the results that we expect from the idea. However, the effectiveness of the results such as the degree of improvements in sampling efficiency and approximation performance will be highly dependent on the answers to the following two questions:

Question 1. How do we determine $\{N_t^i\}_{i=1}^N$ in Step 1?
Question 2. How do we determine $\{\hat{w}_t^j\}_{j=1}^N$ in Step 2?

where N_t^i is the number of times that a sample $x_{0:t}^i$ is selected such that $\sum_{i=1}^N N_t^i = N$.

Since the first introduction, many improved resampling schemes have been presented in the literature with different suggestions for the questions. To our best knowledge, all of the schemes are based on probabilistic approaches to determine $\{N_t^i\}_{i=1}^N$. Accordingly, the weights are reset after resampling (i.e., $\hat{w}_t^j \leftarrow 1/N$ for all j). The two most popular schemes are:

Systematic Resampling [7]: for all $j,\ u_t^j \leftarrow u_t + (j-1)/N$ where $u_t \sim \mathcal{U}(0, \frac{1}{N})$
for all $i,\ N_t^i \leftarrow |\{u_t^j : \sum_{l=1}^{i-1} w_t^l < u_t^j \leq \sum_{l=1}^{i} w_t^l\}|$
for all $j,\ \hat{w}_t^j \leftarrow 1/N$

Residual Resampling [8] : for all $i,\ \bar{N}_t^i \leftarrow \lfloor N w_t^i \rfloor$ and $\bar{w}_t^i \leftarrow N w_t^i - \bar{N}_t^i$
for all $i,\ N_t^i \leftarrow \bar{N}_t^i + \tilde{N}_t^i$
for all $j,\ \hat{w}_t^j \leftarrow 1/N$

where $\{\tilde{N}_t^i\}_{i=1}^N$ are the remainders drawn from a multinomial with $(N - \sum_{i=1}^N \bar{N}_t^i, \{\frac{\bar{w}_t^i}{\sum_{k=1}^N \bar{w}_t^k}\}_{i=1}^N)$ and, $|\cdot|$ and $\lfloor\cdot\rfloor$ denote the cardinality of a set and the floor function, respectively.
Although these probabilistic-based resampling schemes alleviate the sample degeneracy problem by removing trivial samples, they often remove unnecessarily many samples. The next section shows this causes many undesirable side effects, and presents deterministic resampling as an alternative.

3 Deterministic Resampling

3.1 The Basics

Resampling is a practical method to ease the sample degeneracy problem, and should be performed between some SIS iterations. However, we must not use it in the case that the problem does not exist since it only decreases the number of distinctive samples and introduces some extra noise, which inherently result from the process of removing samples [5, 8, 13]. In the practical case that the sample degeneracy problem always exists, such drawbacks of resampling will become worse because of the increase in the number of removed samples. As presented in many of the literature [5, 9–12, 14, 15], the drawbacks bring many serious problems such as the increase of internal correlations and the loss of diversity among samples. Consequently, fully available information will be reduced so that summary statistics will be derived from smaller samples in effect. Furthermore, important regions that samples support will be smaller so that the robustness to outliers will be weakened and the recoverability from a high approximation error accumulated over time will be decreased. However, the most critical problem is that almost samples can be removed after resampling, which causes those serious problems to significantly decrease sampling efficiency and approximation performance. Needless to say, this becomes much critical as the dimension of sample space inceases. Hence, we now have the condition that the answer to the Question 1 should satisfy:

Condition 1. Keep the number of removed samples as small as possible

When the Condition 1 is considered, we can obtain much better results from deterministic-based resampling rather than from probabilistic-based resampling. To see this, suppose we are interested in computing an expectation $E_p(f) = \int f(x)p(x)\,\mathrm{d}x$ where $f(x) = x$, ($x \in \mathbb{R}$ and $1 \leq x \leq 10$), and $p(x)$ is only known up to a normalizing constant. For simple description, $\mathcal{N}(5, 1)$ is selected as $p(x)$ and a discrete uniform distribution is proposed as $q(x)$ where $x \in \{1, 2, \ldots, 10\}$. To estimate the expectation, we now draw 10 samples from $q(x)$ and assign a weight to each sample. If we assume that these samples are perfectly uniformly-distributed, then their weights will be $\{0.0001, 0.0044, 0.0540, 0.2420, 0.3989, 0.2420, 0.0540, 0.0044, 0.0001, 0.0000\}$ in ascending order of sample values. Note that the 10 weighted random samples now very well represent $p(x)$. We then now perform resampling on the samples, with the sample size kept equal after resampling as in general cases. If we use probabilistic-based resampling such as residual or systematic resampling, we can expect that the number of removed samples will be 6, because the probability that the samples having $4, 5$ and 6 are selected is around 0.9. In fact, this result is almost always true because they, especially systematic resampling, are minimum variance resampling schemes. So, we now have only 40% distinct samples. Furthermore, this result will be severe as $p(x)$ is more peaked.

This problem comes from the underlying idea of probabilistic-based resampling for the empirical approximation of $p(x)$. To be more clear, consider the above example again. Right after SIS, we had uniformly distributed random samples that are properly weighted according to $p(x)$ and we saw that these samples very well represent $p(x)$ using the weights. Then after probabilistic-based resampling, we now have unweighted random samples that are properly distributed according to $p(x)$. So, it can be interpreted as the correct weight information is fully used to correct the distribution of samples as if they were drawn from $p(x)$. This is the underlying idea of probabilistic-based resampling [5] and can be good if the sample size increases after resampling. Since that is not acceptable in practice, the underlying idea brings two side effects. First, another approximation error (i.e., MC variation) is added to SMC methods. Second, there always exists a great loss of weight information, which results from the reset of the weights (i.e., equal weights) after resampling. Here, the loss of weight information means any weight information not used to correct the distribution of samples, just like the loss of energy in energy transformation. Clearly, low-weight samples (i.e., samples having low weights) are much weaker against the loss than high-weight samples because even small loss brings the removal of low-weight samples. So, we can reduce the number of removed samples by reducing the loss and the best approach to this is not to reset the weights but keep them as much as possible after resampling. Hence, we now have the condition that the answer to the Question 2 should satisfy:

Condition 2. Keep the loss of weight information as small as possible

Note that, for any resampling scheme, the weights $\{\hat{w}_t^j\}_{j=1}^N$ must be properly assigned such that [8] $\sum_{j=1}^N \hat{w}_t^j f_t(\hat{x}_{0:t}^j)$ approaches (1) as $N \to \infty$. In this sense, the reset of the weights is clearly proper in probabilistic-based resampling. Thus, it is hardly possible to satisfy the Condition 2 using probabilistic approaches. Consequently, any probabilistic-based resampling schemes are hardly good to deal with the Condition 1. As one can easily see, however, deterministic-based resampling can do this easily and better.

3.2 The Method

We now present our proposed scheme, *Deterministic Resampling*. Let us consider again the set of N particles $\{x_{0:t}^i, w_t^i\}_{i=1}^N$ obtained at time t after SIS. At first, our scheme modifies (4) as follows.

$$\sum_{i=1}^{N} w_t^i f_t(x_{0:t}^i) = \sum_{i=1}^{N} \ddot{N}_t^i \frac{w_t^i}{\ddot{N}_t^i} f_t(x_{0:t}^i) \qquad \text{where} \qquad \ddot{N}_t^i = \left\lceil \frac{w_t^i}{(2/N)} \right\rceil \tag{5}$$

where $\lceil \cdot \rceil$ denotes the ceiling function. Then, our scheme reads the modified one as follows.

there are $\ddot{N}_t^i$ copies of a sample $x_{0:t}^i$ whose weight is $w_t^i/\ddot{N}_t^i$, where $\ddot{N}_t^i$ is determined by $\lceil w_t^i/(\frac{2}{N}) \rceil$

So, if $w_t^i \leq \frac{2}{N}$, the number of the sample $x_{0:t}^i$ is just one, otherwise, more than one.

Note that we change nothing on (4) so far, but just modify it and read it differently. However, this brings two advantages if we set $N_t^i \leftarrow \ddot{N}_t^i$ in the Question 1 and set $\hat{w}_t^j \leftarrow w_t^i/\ddot{N}_t^i$ when $\hat{x}_{0:t}^j \leftarrow x_{0:t}^i$ in the Question 2. First, no additional approximation error is added to SMC methods because there is no variation in (4). Second, no loss of weight information exists because every weight information is kept either in one sample or several samples.

Altough this method is not practical due to the increase of the sample size after resampling, we can expect to obtain the best results by removing minimum-weight samples. That is, we can minimize the additional approximation error and the loss of weight information by removing $(\sum_{i=1}^{N} \ddot{N}_t^i - N)$ minimum-weight samples after the method. Then clearly, $\sum_{i=1}^{N} N_t^i = N$ after resampling. Furthermore, this reduces the effects of the sample degeneracy problem because this follows exactly what the basic idea of resampling pursue.

One more advantage of the modified form comes from the term $(\frac{2}{N})$, which we call as *cutoff weight*. This cutoff weight plays a role as the maximum weight. That is, after resampling, no sample will have a weight larger than $(\frac{2}{N})$. Also, this cutoff weight plays a role as the threshold weight. During resampling, only a sample whose weight is larger than $(\frac{2}{N})$ will be allowed to be duplicated. This has two benefits. First, we make one sample has a weight between zero and $2/N$, which is beneficial because this makes samples distributed according to a target density. This can be understood in the same context as the equal weights (i.e., $1/N$) in the ideal case (i.e., $q_t(x_{0:t}) = p_t(x_{0:t})$) and the unbiasedness (i.e., $\mathbb{E}[N_t^i | \{w_t^i\}] = N w_t^i$) [18] in probabilistic-based resampling. Also, we limit the number of removed samples to $N/2$.

Proposition 1. *The number of removed samples never exceeds* $\frac{N}{2}$.

Proof: Let N_t^r be the number of removed samples after deterministic resampling at time t.
1. Then, $N_t^r = \sum_{i=1}^{N} \ddot{N}_t^i - N = \sum_{i=1}^{N} \lceil N w_t^i/2 \rceil - N$.
2. By the definition of the ceiling function, it is true that $\lceil N w_t^i/2 \rceil - 1 < N w_t^i/2$ for all i.
3. Thus, $\sum_{i=1}^{N} (\lceil N w_t^i/2 \rceil - 1) < \sum_{i=1}^{N} N w_t^i/2 \Longrightarrow \sum_{i=1}^{N} (\lceil N w_t^i/2 \rceil) - N < N/2 \sum_{i=1}^{N} w_t^i$.
4. Since $\sum_{i=1}^{N} w_t^i = 1$, we show that $N_t^r < N/2$.

This property shows that deterministic resampling never removes more than half samples even in a case that the sample degeneracy problem is very severe, while other resampling schemes remove all samples except one. This prevents the drawbacks and problems presented in 3.1 becoming serious. So, we can expect significant improvements in sampling efficiency and approximation performance compared to current resampling schemes. To summarize, we now answer the Question 1 and 2.

Deterministic Resampling : for all i, $\ddot{N}_t^i \leftarrow \lceil w_t^i/(\frac{2}{N}) \rceil$ and $\ddot{w}_t^i \leftarrow w_t^i/\ddot{N}_t^i$
for all i, $N_t^i \leftarrow \ddot{N}_t^i - \lceil \bar{w}_t - \ddot{w}_t^i \rceil$
for all j, $\hat{w}_t^j \leftarrow \ddot{w}_t^i$ if $\hat{x}_{0:t}^j \leftarrow x_{0:t}^i$ for any i

where $\bar{w}_t$ is the $(\sum_{i=1}^{N} \ddot{N}_t^i - N + 1)^{\text{th}}$ smallest weight among $\{\ddot{w}_t^i\}_{i=1}^{N}$. We show $\sum_{i=1}^{N} N_t^i = N$.

Proposition 2. $\sum_{i=1}^{N} \lceil \bar{w}_t - \ddot{w}_t^i \rceil = \sum_{i=1}^{N} \ddot{N}_t^i - N$

Proof: If we assume $\bar{w}_t > \bar{\bar{w}}_t$, where $\bar{\bar{w}}_t$ is the $(\sum_{i=1}^{N} \ddot{N}_t^i - N)^{\text{th}}$ smallest weight among $\{\ddot{w}_t^i\}_{i=1}^{N}$, Proposition 2 is true because it is true, by Proposition 1, that any removed sample $x_{0:t}^i$ had $\ddot{N}_t^i = 1$.

Therefore, our answers are correct with the assumption. Using our scheme for the example presented in 3.1, one can easily see that the number of removed samples are always 3. So, we have 70% distinct samples. Also, we can see that the loss of weight information is only 0.0002, which is very small compared with those of systematic and residual resampling schemes. We now end this subsection with one more property of our scheme without proof because the proof is straightforward by (5).

Proposition 3. *After deterministic resampling,* $\sum_{j=1}^{N} \hat{w}_t^j f_t(\hat{x}_{0:t}^j)$ *approaches* (1) *as* $N \to \infty$.

3.3 The Algorithm

The deterministic resampling algorithm is given in Algorithm 1. This algorithm requires a sorting method (e.g., quicksort) to select N largest-weight samples. Thus, the expected running time of this algorithm is $O(N \lg N)$ [19], whereas systematic and residual resampling generally take $O(N)$ time. In practice, this difference is trivial if N is not very large depending on computing power. However, it becomes serious when we have very large N running on a device having limited computing power. In this case, a linear-time deterministic resampling algorithm is necessary. Such an algorithm can be implemented using a selection method [19] instead a sorting method. In fact, our scheme does not

Algorithm 1 Deterministic Resampling

```
Input  : $\{x_{0:t}^i, w_t^i\}_{i=1}^N$
Output : $\{\hat{x}_{0:t}^j, \hat{w}_t^j\}_{j=1}^N$
for $i \leftarrow 1$ to $N$ do
    $N_t^i \leftarrow \lceil N w_t^i / 2 \rceil$
    $w_t^i \leftarrow w_t^i / N_t^i$
$index \leftarrow$ sort $\{w_t^i\}_{i=1}^N$ in descending order, then return the weight indices of the sorted array
$k \leftarrow 1$
$weighttotal \leftarrow 0$
for $j \leftarrow 1$ to $N$ do
    $i \leftarrow index(k)$
    $\{\hat{x}_{0:t}^j, \hat{w}_t^j\} \leftarrow \{x_{0:t}^i, w_t^i\}$
    $weighttotal \leftarrow weighttotal + \hat{w}_t^j$
    $N_t^i \leftarrow N_t^i - 1$
    if $N_t^i == 0$ then $k \leftarrow k + 1$
for $j \leftarrow 1$ to $N$ do $\hat{w}_t^j \leftarrow \hat{w}_t^j / weighttotal$        /* normalization */
```

require $\{w_t^i\}_{i=1}^N$ to be fully sorted. As described in 3.2, it is enough to find the $(\sum_{i=1}^N N_t^i - N + 1)^{\text{th}}$ smallest weight in $\{w_t^i\}_{i=1}^N$ after the first **for** loop in Algorithm 1, and then use the ceiling function with special care of the assumption presented in the proof of Proposition 2, to obtain $\sum_{i=1}^N N_t^i = N$. Since this selection problem can be solved in worst-case linear time [19], we can implement deterministic resampling which runs in $O(N)$ in the worst case.

However, one might prefer the sorting-based algorithm for practical use due to the implementation issue. Since there now exist some powerful sorting algorithms such as quicksort which are very easy to implement, the sorting-based algorithm presented in Algorithm 1 is also easy to implement even compared to systematic resampling.

4 Simulation Study

We now illustrate the performance of our scheme with two well-known recursive Bayesian estimation problems: Univariate Nonstationary Growth Model [2, 12] and Bearings-only Tracking [9, 10]. These problems recursively require estimation of unknown state x_t from a sequence of noisy and partial observations $z_{1:t}$; see [12, 16, 20] for more details on the recursive Bayesian estimation.

4.1 Univariate Nonstationary Growth Model

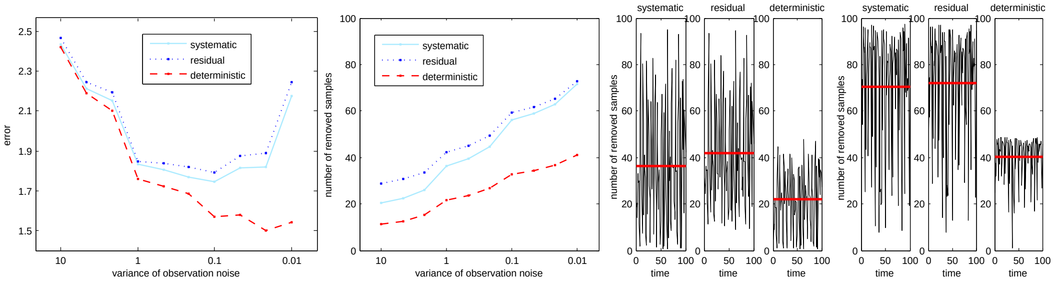
Consider the following dynamic nonlinear state space model as described in [2, 12]

$$x_t = 0.5x_{t-1} + \frac{25x_{t-1}}{(1 + x_{t-1}^2)} + 8\cos(1.2t) + v_t$$

$$z_t = \frac{x_t^2}{20} + n_t$$

where v_t and n_t are zero mean Gaussian system noise with variance Q_t and observation noise with variance R_t, respectively. We assume that true states evolve in 100 time steps from a known initial state $x_0 = 0$. To estimate the states, we use Bootstrap Filter [2] of the sample size $N = 100$ but with 3 different resampling schemes: systematic, residual, and deterministic. Figure 1 shows the results.

Figure 1 (a) shows the estimation error against 10 different R_t from 10 to 0.01 with fixed $Q_t = 1$. Note that smaller R_t compared to Q_t indicates more severe sample degeneracy problem because the Bootstrap filter draws samples from prior but assigns weights by likelihood. So, the variance of weights will increase as R_t decreases at fixed Q_t. In this case, the number of removed samples will increase and there will be only few distinct samples if we use systematic and residual resampling as we described in 3.1. Let us first consider the case that R_t is larger than Q_t. The figure shows that



(a) (b) (c) $Q_t = 1, R_t = 1$ (d) $Q_t = 1, R_t = 0.01$

Figure 1: Performance comparison between three resampling schemes through univariate nonstationary growth model. (a) Error is measured by root mean square error (RMSE) in 100 MC runs and averaged over time. (b) The number of removed samples is averaged over time at each R_t. (c-d) A thick red straight line shows the average number of removed samples at each scheme.

the performance improvement is not significant in this case. This is because the sample degeneracy problem is so trivial that all schemes neither remove many samples nor have the great loss of weight information in this case. As R_t decreases, however, our scheme shows significant performance improvements on systematic and residual as we expected. Especially, when we have small observation noise, the error in our scheme still decreases whereas the performance of the others is degraded. This implies that more powerful observation sensors give higher performance in our scheme. Note that these sensors reduce performance as described in [21] when we use systematic and residual. This significant improvements mainly come from the small number of removed samples as shown in Figure 1 (b), (c), and (d). Especially for the small observation noise in Figure 1 (d), we can see that our scheme never removes more than half samples while the others remove almost all samples in many time steps. As described in 3.1, this large number of removed samples causes SMC methods to behave as if they have very small sample size N, which also wastes a large computational effort.

4.2 Bearings-only Tracking

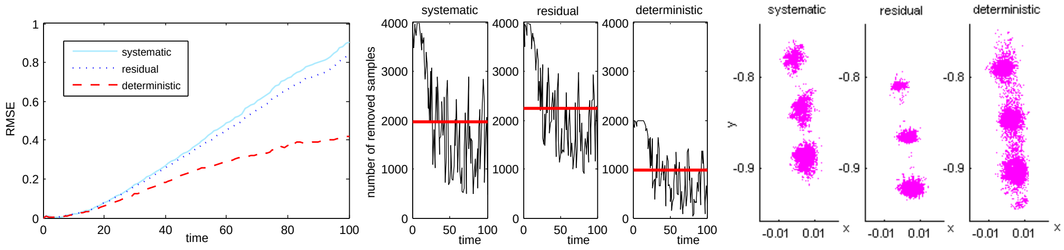
In this problem, one target moves in a scene and one observer tracks the target by sensing the target bearings in the Cartesian coordinates. The target state $\mathbf{x}_t$ at time t includes the horizontal position x_t and velocity $\dot{x}_t$, and the vertical position y_t and velocity $\dot{y}_t$ so that $\mathbf{x}_t = (x_t\ \dot{x}_t\ y_t\ \dot{y}_t)'$. The observer at the origin takes a noisy observation z_t at time t. Then, this problem can be modeled as follows.

$$\mathbf{x}_t = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_{t-1} + \begin{pmatrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{pmatrix} \mathbf{v}_t$$

$$z_t = \tan^{-1}(y_t/x_t) + n_t$$

where $\mathbf{v}_t = (v_x\ v_y)'_t$ and n_t are zero mean Gaussian system noise with covariance: $0.01^2 \times \mathbf{I}_2$ (where $\mathbf{I}_2$ is the 2-by-2 identity matrix) and observation noise with variance: 0.005^2, respectively. The true initial state $\mathbf{x}_0$ is $x_0 = -0.05$, $\dot{x}_0 = 0.001$, $y_0 = 0.2$, $\dot{y}_0 = -0.055$. We assume this initial state is known as a Gaussian distribution with true mean (i.e., mean $= \mathbf{x}_0$) and diagonal covariance of the elements $0.01 \times \text{diag}(0.5^2, 0.005^2, 0.3^2, 0.01^2)$. In this problem, Auxiliary Particle Filter [9] is used but with only one resampling stage at each time step [5]. We also compare 3 resampling schemes just like 4.1 but with $N = 4000$. We now discuss our results shown in Figure 2 and 3.

Figure 2 (a) shows the estimation error measured by RMSE in 100 MC runs. We can see that our scheme outperforms the others in terms of estimation accuracy. However, the interesting point is that our scheme always removes fewer samples than the others as shown in Figure 2 (b). This implies that our scheme has much broader and rich diversity of samples compared to the others. We show this in Figure 2 (c). These results say that our scheme estimates the true states better than the others using the equal number of samples which support much broader state spaces than the others. So, our scheme improves sampling efficiency and approximation performance. We expect that this comes



(a) (b) (c)

Figure 2: Performance comparison between three resampling schemes through bearings-only tracking over 100 time steps. (b) A thick red straight line shows the average number of removed samples at each scheme. (c) At each sub-figure, we can see three clusters. The upper cluster was obtained by drawing a scatter plot of the x and y positions of all samples at time $t = 18$, where each one is marked by magenta point. So, the points show the coverage area that 4000 samples support to estimate the target location at time $t = 18$. Then, the middle cluster was obtained as the same way at time $t = 19$ and the lower cluster was at time $t = 20$.

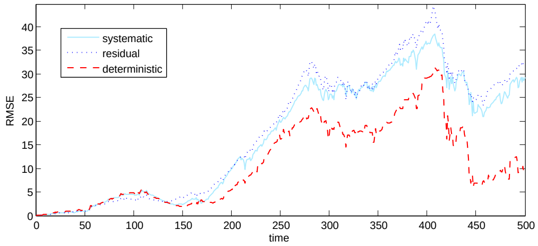


Figure 3: Performance comparison between three resampling schemes through bearings-only tracking over 500 time steps.

from the small loss of weight information which implies that we have more informative samples using the weight information although this is not explicitly shown in this filter due to the exchange of SIS and resampling.

We now evaluate the 3 resampling schemes through a more challenging task. We run this problem over 500 time steps. Since the correction of the errors only relies on the observation of the target bearings, once the estimation fails to track the target, then it is very hard to recover from the error if no samples exist on the true regions. This is a quite challenging task because small error will be accumulated over time and there is no additional sensor such as GPS to fix this accumulated error. Figure 3 shows the estimation error measured by RMSE in 100 MC runs. This result shows that our scheme significantly improves the estimation errors compared to the other resampling schemes as we expected.

5 Conclusion

In this paper, we proposed a new resampling scheme designed to not only alleviate the sample degeneracy problem but also reduce the number of removed samples by applying a completely deterministic approach to resampling and keep the weight information properly. Through analysis and two simulation studies, we showed that the proposed scheme never removes samples more than half of the sample size in any case and this improves sampling efficiency and approximation performance. Although the performance of the three resampling schemes are almost similar when the sample degeneracy problem is trivial, there will be large benefits from our scheme once the problem is severe in many applications.

References

[1] Liu, J.S. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag, New York.

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

[3] Andrieu, C., de Freitas, N., Doucet, A. and Jordan, M.I. (2003). An Introduction to MCMC for Machine Learning. *Machine Learning*, **50**, 5-43.

[4] Kong, A., Liu, J.S. and Wong, W.H. (1994). Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association*, **89**, 278-288.

[5] Doucet, A. and Johansen, A.M. (2009). A Tutorial on Particle Filtering and Smoothing: Fifteen years later. *The Oxford Handbook of Nonlinear Filtering* (eds. Crisan, D. & Rozovsky, B.), Oxford University Press.

[6] Liu, J.S. and Chen, R. (1995). Blind Deconvolution via Sequential Imputations. *Journal of the American Statistical Association*, **90**, 567-576.

[7] Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, **5**, 1-25.

[8] Liu, J.S. and Chen, R. (1998). Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, **93**, 1032-1044.

[9] Pitt, M.K. and Shephard, N. (1999). Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, **94**, 590-599.

[10] Carpenter, J., Clifford, P. and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation, IEE Proceedings*, **146**, 2-7.

[11] Fearnhead, P. (2002). Markov Chain Monte Carlo, Sufficient Statistics, and Particle Filters. *Journal of Computational and Graphical Statistics*, **11**, 848-862.

[12] Arulampalam, M.S., Maskell, S., Gordon, N. and Clapp, T. (2002). A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *Signal Processing, IEEE Transactions on*, **50**, 174-188.

[13] Chopin, N. (2004). Central Limit Theorem for Sequential Monte Carlo Methods and Its Application to Bayesian Inference. *The Annals of Statistics*, **32**, 2385-2411.

[14] Kunsch, H.R. (2005). Recursive Monte Carlo Filters: Algorithms and Theoretical Analysis. *The Annals of Statistics*, **33**, 1983-2021.

[15] Doucet, A., Briers, M. and Senecal, S. (2006). Efficient Block Sampling Strategies for Sequential Monte Carlo Methods. *Journal of Computational and Graphical Statistics*, **15**, 693-711.

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

[17] Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, **7**, 1-26.

[18] Douc, R., Cappe, O. and Moulines, E. (2005). Comparison of Resampling Schemes for Particle Filtering. *Image and Signal Processing and Analysis, Proceedings of the 4th International Symposium on*, 64-69.

[19] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2001). *Introduction to Algorithms, second edition*. MIT Press.

[20] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R. and Nordlund, P. (2002). Particle Filters for Positioning, Navigation, and Tracking. *Signal Processing, IEEE Transactions on*, **50**, 425-437.

[21] Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence*, **128**, 99-141.