# Comparing Resampling Techniques for Multitarget Tracking using Particle Filtering

Henrik Karlsson
henrik10@kth.se

January 18, 2018

**Abstract**

Lorem ipsum

## 1 Introduction

The Particle Filter (PF), also known as Sequential Importance Sampling and Resampling (SISR), is a Monte Carlo, or simulation based algorithm, for recursive Bayesian inference [2]. The PF consists of particles and associated importance weights that are propagated through time to approximated a target distribution. It only needs a proposal distribution, a likelihood and a dynamic model. The PF is used in many areas such as tracking, parameter estimation, robotics, etc.

The PF is an improvement over the Sequential Importance Sampling (SIS) [2]. SIS have the problem of degeneracy; that is, after a few iterations, most of the particles will have negligible weight. The PF improves upon SIS by adding the resampling step where particles with low weight are eliminated and replaced by copies of the surviving particles. More specifically, the new set $\{\hat{z}_t^s\}_{s=1}^S$ is sampled from the distribution

$$p(z_t|y_{1:t}) \approx \sum_{s=1}^{S} w_t^s \delta_{z_t^s}(z_t).$$

However, this leads to another problem, particle deprivation.

Particle deprivation is when the particles do not cover regions of high probability [3], this is a significant problem of PF. This generally happens when the number of particles is not large enough and/or the target distribution is multi-modal. Particle deprivation occurs due to the sampling variance and thus the resampling step can wipe out all particles in the high density areas of the target distribution. The probability of this happening is non-zero at each re-sampling step and therefore it is only a matter of time until it happens. Solutions to particle deprivation is to add more particles, to randomly generated particles in each iteration, or use a better sampler.

Multitarget tracking (MTT) is the localization and recursive detection of objects of interest based on sequential measurements. Some examples are aircraft tracking using radar, and tracking people through a video feed. In practice, there are many factors that contributes to uncertainty of an objects location such as noise in measurement, clutter and environment. Therefore, a probabilistic approach to the problem is required. Popular approaches are Bayesian Monte Carlo Estimation such as particle filtering.

Particle filters have some problems with multitarget tracking. Due to that MTT problems are multi-modal, PF solutions tends to suffer from particle deprivation and will therefore lose targets. One of the solutions to this problem is use a sampler specifically made to track multiple targets.

This paper compares different resampling techniques for PF in the context of multi-target tracking.

# 2 Related Work

# 3 Resampling methods

## 3.1 Systematic Resampling

Systematic resampling is widely used resampling method for particle filters. It is preferred because it is computationally simple and have good empirical performance [1]. The systematic resampling method have shown to be empirically comparable with other resampling methods such as stratified sampling and residual resampling which in turn have been shown to be better than multinomial resampling [1].

In practice it is implemented as follows:

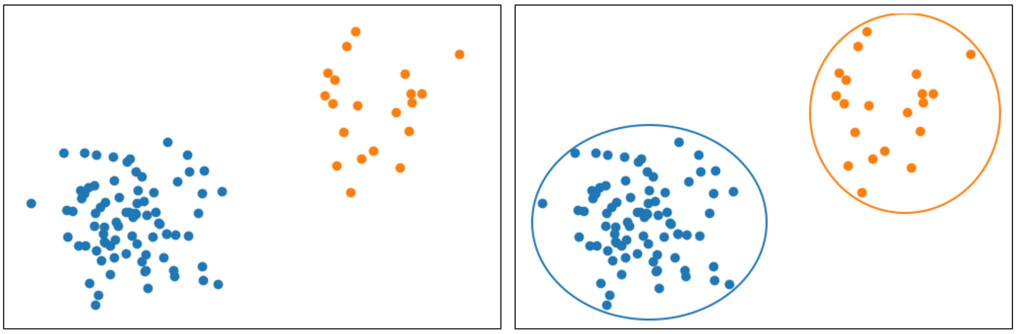
**Algorithm 1** Systematic Resampling algorithm

Draw: $r \sim U(0,1)$
**for** $i = 0 : M - 1$ **do**
  $U^i \leftarrow (i + r)/M$
  $I^i \leftarrow D_w^{inv}(U_i)$
**end for**

Where $D_w^{inv}$ is the inverse of the cumulative distribution function associated with the particle weights $\{w_t^i\}_{i=1}^N$, $M$ is the number of samples to draw, and $I^i$ is the index of the $i$'th sample. This resampling method is sensitive to the order of the particles before the resampling as it changes the cumulative distribution function.

## 3.2 Cluster Resampling

The idea behind cluster resampling is that we are not interested in particles, we are interested in the high probability areas the particles represent. One of the main problems of systematic resampling is that sampling variance can eliminate all particles in one area (figure 1a). Cluster resampling remedies this problem by representing areas as a cluster/mixture of particles and then resample particles from one cluster at the time. The number of particles resampled from one cluster is proportional to the weight of the cluster, thus if an cluster have 10% of the weight, we draw 10% of the particles from that cluster (figure 1b).



(a) Standard resampling from this distribution can eliminate the all orange particles due to sampling variance since there is nothing preventing the resampler from only selecting blue particles.

(b) By clustering particles and thereafter resampling 10% of the particles from the orange cluster and 90% from the blue cluster, there will be exactly 10 orange and 90 blue particles in the new set.

Figure 1: 100 hundred particles to be resampled. 90% of the weight is in the blue particles and 10% is in the orange particles.

#### 3.2.1 Algorithm

The generic algorithm for ones step of cluster resampling is the following:

**step 1:** Propagate particles

**step 2:** Weight update

**step 3:** Resample particles

**step 4:** Calculate particles clusters

where the step 3 and 4 can exchange place.

#### 3.2.2 Mixture Representation

In cluster resampling, we represent the target distribution as an $M$-component mixture model. Let $z_t$ denote the state at time $t$ and let $y^t = \{y_1, y_2, \cdots, y_t\}$ be the observations. Then the target distribution $p(z_t|y^t)$ is the following,

$$p(z_t|y^t) = \sum_{m=1}^{M} \pi_{m,t} p_m(x_t|y^t) \tag{1}$$

where $\sum_{m=1}^{M} \pi_{m,t} = 1$.

The particle filter approximation of the mixture model represents each mixture as a cluster of particles. Let the particles $\{z_t^i\}_{n=1}^{N}$ be clustered into $M$ different clusters $\mathcal{I}_{m,t}$ representing the mixture components $m$. Let each particle $z_t^i$ belong to one and only one cluster $m$ and let $c_{t,i} = m$. The mixture components $p_m(z_t|y^t)$ are approximated by:

$$q_m(z_t|y^t) = \sum_{i \in \mathcal{I}_{m,t}} w_t^i \delta_{z_t^i}(z_t) \tag{2}$$

Inserting (2) into (1) gives us the following approximation of $p(z_t|y^{t-1})$

$$q(z_t|y^{t-1}) = \sum_{m=1}^{M} \pi_{m,t} \sum_{i \in \mathcal{I}_{m,t}} w_t^i \delta_{z_t^i}(z_t) \tag{3}$$

where the weights $\pi_{m,t}$ and $w_i^t$ are computed as follows:

$$\pi_{m,t} = \frac{\hat{\pi}_{m,t}}{\sum_{m'=1}^{M} \hat{\pi}_{m',t}}, \; w_t^i = \frac{\hat{w}_t^i}{\hat{\pi}_{c_i,t}} \tag{4}$$

$$\hat{\pi}_{m,t} = \sum_{i \in \mathcal{I}_{m,t}} \hat{w}_t^i, \; \hat{w}_t^i = p(y_t|z_t^i) w_{t-1}^i. \tag{5}$$

It can be shown that this approximation is identical to the approximation used in normal particle filtering. For a complete derivation, see [4].

### 3.2.3 Resampling step

The main difference between standard resampling methods (e.g. systematic resampling), and the cluster resampling is that we draw samples from mixture components. When drawing samples from one mixture, we can use standard resampling methods such as systematic resampling.

When drawing $n$ samples from component $m$, we have to make sure that sum of the weights of particles in $m$ have to be $\pi_{m,t}$. Thus, the particle weights have to be $\frac{\pi_{m,t}}{n}$. Let $N$ be the total number of particles drawn and let $n = \pi_{m,t} N$; that is, $n$ is proportional to the weight $\pi_{m,t}$. Then the new particle weights are $\frac{1}{N}$. For a complete derivation, see [4].

### 3.2.4 Clustering step

Clustering is the task to group objects in such a way that objects in the same group are more similar to each other than other groups. The algorithms we are interested in are two versions of the k-means clustering algorithms, Lloyd's algorithm and the EM algorithm.

**Lloyd's algorithm** Given an initial set of $M$ centroids $m_1, m_2, \ldots, m_M$, the k-means algorithm proceeds in two steps:

> **Assignment:** Every object $z^i$ is assigned to the nearest centroid $k$. The distance is measured with the euclidean distance.
>
> **Update:** New centroids $m'_k$ are calculated by taking the mean of each object $z^i$ that belongs to $k$; that is, $m'_k = \frac{1}{|\mathcal{I}_k|} \sum_{i \in \mathcal{I}_k} z^i$, where $\mathcal{I}_k$ is the set of objects assigned to centroid $k$.

this is done iteratively until the assignments no longer changes.

**EM algorithm** The algorithm starts with an initial initial set of mean vectors $m_1, m_2, \ldots, m_M$. Then the EM algorithm proceeds in 2 steps:

> **Expectation:** Every object $i$ is partially assigned to each mixture $k$, this is represented by a weight $w_k^i = \frac{p_k(z^i)}{\sum_{k'=1}^{M} p_{k'}(z^i)}$ where $p_k(z^i) = \mathcal{N}(z^i|m_k, \sigma^2 I)$.

**Maximization:** New means $m'_k$ are calculated by taking the weighted mean of each object $z^i$; that is, $m'_k = \frac{\sum_{i \in \mathcal{I}_k} w^i_k z^i}{\sum_{i \in \mathcal{I}_k} w^i_k}$.

this is repeated until convergence. The EM algorithm for clustering is based on the EM algorithm for GMM.

Both algorithms suffers from local minima; that is, the optimal clustering is not found. Because of the local minima problem, the algorithms may need to be run several times to find a better clustering. They algorithms are also biased towards spherical or hyper-spherical clusters. The EM algorithm is in general better than Lloyd's algorithm since it has fewer local minimas, however, Lloyd's algorithm is easier to implement and faster.

### 3.3 Parallel Particle Filter

Clustering is the task to group objects in such a way that objects in the same group are more similar to each other than other groups. The algorithms we are interested in are the k-means algorithm and the EM algorithm for clustering since they are computationally fast relative to other algorithms.

## 4 Experimental Results

## 5 Summary and Conclusions

## References

[1] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pages 64–69. IEEE, 2005.

[2] Kevin P. Murphy. *Machine Learning, a Probabilistic Perspective.* The MIT Press, 2012.

[3] Wolfram Burgard Sebastian Thrun and Dieter Fox. *Probabilistic robotics.* The MIT Press, 2006.

[4] Jaco Vermaak, Arnaud Doucet, Perez Patrick, et al. Maintaining multi-modality through mixture tracking. In *null*, page 1110. IEEE, 2003.