

Residual resampling in SMC

Suzie Brown

February 1, 2019

intro

state multinomial resampling procedure for comparison

Assuming constant number of particles and unbiasedness of resampling scheme...

Also define N, w's, v's, etc.

1 Low variance resampling

It is well known that multinomial resampling does not yield the most efficient sequential Monte Carlo algorithm. A resampling step is necessary in SMC to prevent the variance of the particle weights to explode (*weight degeneracy*), and it must incorporate some randomness to ensure convergence to the target distribution. However, the addition of a random resampling procedure causes the ancestral lines of the particles to coalesce back in time (*ancestral degeneracy*). If we are interested in estimating the smoothing distribution, we have to trade off these two effects: roughly speaking, less random resampling leads to more weight degeneracy and less ancestral degeneracy.

As long as the resampling procedure is able to kill off less promising particles reasonably efficiently, there shouldn't be an issue with weight degeneracy. For this reason, a lot of work has gone into suggesting improvements to multinomial resampling which reduce the speed of coalescence whilst still allowing 'bad' particles to be killed off quickly. One such scheme, particularly popular among practitioners, is *residual resampling* (Liu and Chen, 1998). The offspring counts are sampled according to:

$$v_t^{(i)} = \lfloor Nw_t^{(i)} \rfloor + X_i$$
$$X_i \sim \text{Multinomial}(N - k, (\bar{w}_t^{(1)}, \dots, \bar{w}_t^{(N)}))$$

where $k := \sum_{i=1}^N \lfloor Nw_t^{(i)} \rfloor$ is the number of offspring assigned deterministically, and $\bar{w}_t^{(i)} := \frac{Nw_t^{(i)} - \lfloor Nw_t^{(i)} \rfloor}{N - k}$ are the residual weights. Let us also define the residuals $r_i := Nw_t^{(i)} - \lfloor Nw_t^{(i)} \rfloor$.

We can see that this scheme offers a reduction in the variance of the weights (compared to no resampling): particles with weight less than $1/N$ stand the chance of having no offspring and therefore being killed off, while any particle with weight at least $1/N$ is guaranteed to survive. As in multinomial resampling, particles with very low weights are very unlikely to survive.

Furthermore, Douc and Cappé (2005) show that the conditional variance of the resulting approximation is always smaller with residual resampling than with multinomial resampling. They note that the same applies to stratified resampling, but the two schemes perform similarly in practice.

The variance of offspring counts among a particular generation serves as an intuitive proxy for the amount of ancestral degeneracy associated with a particular resampling scheme. To illustrate this, consider two extremes: a resampling scheme that assigns all offspring to the single highest-weighted particle, versus a scheme that deterministically assigns exactly one offspring to each particle. The first scheme maximises the variance of offspring counts, and causes all the particles' ancestries to coalesce just one iteration back in time. The second, which is the same as no resampling at all, gives zero variance of offspring counts, and the resulting ancestries will never coalesce.

We can see residual resampling (as well as other popular schemes like systematic resampling and stratified resampling) as a modification of multinomial resampling that reduces the amount of randomness in the resampling step. The number of offspring assigned to each particle is decomposed into a deterministic and a random part, as opposed to multinomial resampling which is purely random.

2 Coalescence rate

We shall calculate the expected coalescence rate in the case of residual resampling. The coalescence rate is defined in Koskela et al. (2018) as

$$c_N(t) := \frac{1}{(N)_2} \sum_{i=1}^N \mathbb{E} \left[(v_t^{(i)})_2 \right].$$

The inner expectation comes out as

$$\begin{aligned} \mathbb{E}[(v_t^{(i)})_2] &= \mathbb{E}[(v_t^{(i)})^2] - \mathbb{E}[v_t^{(i)}] \\ &= \lfloor Nw_t^{(i)} \rfloor^2 + 2\lfloor Nw_t^{(i)} \rfloor r_i + r_i \left(1 - \frac{r_i}{N-k} + r_i \right) - Nw_t^{(i)} \\ &= \lfloor Nw_t^{(i)} \rfloor^2 - \lfloor Nw_t^{(i)} \rfloor + 2\lfloor Nw_t^{(i)} \rfloor r_i + r_i^2 \left(1 - \frac{1}{N-k} \right) \\ &= (Nw_t^{(i)})^2 - \lfloor Nw_t^{(i)} \rfloor - \frac{r_i^2}{N-k} \end{aligned}$$

so we get

$$\begin{aligned} \mathbb{E}[c_N^r(t)|\mathcal{F}_{t-1}] &= \frac{1}{(N)_2} \mathbb{E} \left[\sum_{i=1}^N \mathbb{E}[(v_t^{(i)})_2] | \mathcal{F}_{t-1} \right] \\ &= \frac{N}{N-1} \sum_{i=1}^N \mathbb{E}[(w_t^{(i)})^2 | \mathcal{F}_{t-1}] - \frac{1}{(N)_2} \sum_{i=1}^N \mathbb{E} \left[\frac{r_i^2}{N-k} | \mathcal{F}_{t-1} \right] - \frac{1}{(N)_2} \mathbb{E}[k | \mathcal{F}_{t-1}] \\ &= \mathbb{E}[c_N^m(t) | \mathcal{F}_{t-1}] \left(1 + \frac{1}{N-1} \right) - \frac{1}{(N)_2} \mathbb{E} \left[\frac{\sum_{i=1}^N (Nw_t^{(i)} - \lfloor Nw_t^{(i)} \rfloor)^2}{\sum_{j=1}^N (Nw_t^{(j)} - \lfloor Nw_t^{(j)} \rfloor)} | \mathcal{F}_{t-1} \right] \\ &\quad - \frac{1}{(N)_2} \mathbb{E} \left[\sum_{i=1}^N \lfloor Nw_t^{(i)} \rfloor | \mathcal{F}_{t-1} \right] \end{aligned}$$

Sanity check:

When the weights are all equal, $w_t^{(i)} \equiv 1/N$, we should have $\mathbb{E}[c_N^r(t) | \mathcal{F}_{t-1}] = 0$ since each particle will have exactly one offspring so it is impossible for any lineages to coalesce. In this case we have $\mathbb{E}[c_N^m(t) | \mathcal{F}_{t-1}] = \sum_{i=1}^N \mathbb{E}[(w_t^{(i)})^2 | \mathcal{F}_{t-1}] = 1/N$ for multinomial resampling. We also have that $Nw_t^{(i)} \equiv \lfloor Nw_t^{(i)} \rfloor \equiv 1$ and hence $r_i = 0$ and $k = N$. Thus the RHS comes out as

$$\frac{1}{N} \frac{N}{N-1} - 0 - \frac{1}{(N)_2} N = \frac{1}{N-1} - \frac{1}{N-1} = 0$$

as expected.

References

- Douc, R. and Cappé, O. (2005), Comparison of resampling schemes for particle filtering, *in* ‘Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on’, IEEE, pp. 64–69.
- Koskela, J., Jenkins, P. A., Johansen, A. M. and Spano, D. (2018), ‘Asymptotic genealogies of interacting particle systems with an application to sequential monte carlo’, *arXiv preprint arXiv:1804.01811*.
- Liu, J. S. and Chen, R. (1998), ‘Sequential monte carlo methods for dynamic systems’, *Journal of the American statistical association* **93**(443), 1032–1044.