

# ADPF for overdispersed continuous-time Markov chains

July 9, 2024

## Abstract

We propose a combination of MOP and DOP to carry out inference for overdispersed continuous time Markov chains. This algorithm, currently called PPOP (Proposal Partially Off Policy), is described in Section 3. First, we review MOP and DOP.

## 1 MOP- $\alpha$

---

### Algorithm 1: Measurement off policy (MOP- $\alpha$ ) SMC:

---

- 1 Initialize filter particles: simulate  $X_{0,j}^{F,\theta} \sim f_{X_0}(\cdot; \theta)$  for  $j$  in  $1:J$
  - 2 Initialize relative weights:  $w_{0,j}^{F,\theta} = 1$  for  $j$  in  $1:J$
  - 3 **for**  $n$  in  $1:N$  **do**
  - 4     Simulate for prediction:  $X_{n,j}^{P,\theta} \sim f_{X_n|X_{n-1}}(\cdot | X_{n-1,j}^F; \theta)$  for  $j$  in  $1:J$
  - 5     Prediction weights with discounting:  $w_{n,j}^{P,\theta} = (w_{n-1,j}^{F,\theta})^\alpha$  for  $j$  in  $1:J$
  - 6     Evaluate measurement density:  $g_{n,j}^\theta = f_{Y_n|X_n}(g_n^* | X_{n,j}^{P,\theta}; \theta)$  for  $j$  in  $1:J$
  - 7     Before-resampling conditional likelihood:  $L_n^{B,\theta,\alpha} = \frac{\sum_{j=1}^J g_{n,j}^\theta w_{n,j}^{P,\theta}}{\sum_{j=1}^J w_{n,j}^{P,\theta}}$
  - 8     Conditional likelihood under  $\phi$ :  $L_n^\phi = \frac{1}{J} \sum_{m=1}^J g_{n,m}^\phi$
  - 9     Normalize weights:  $\tilde{g}_{n,j}^\phi = \frac{g_{n,j}^\phi}{J L_n^\phi}$  for  $j$  in  $1:J$
  - 10    Apply systematic resampling to select indices  $k_{1:J}$  with  $\text{Prob}(k_j = m) = \tilde{g}_{n,m}^\phi$
  - 11    Resample particles:  $X_{n,j}^{F,\theta} = X_{n,k_j}^{P,\theta}$
  - 12    Filter weights corrected for resampling:  $w_{n,j}^{FC,\theta} = w_{n,j}^{P,\theta} \times \frac{g_{n,j}^\theta}{g_{n,j}^\phi}$  for  $j$  in  $1:J$
  - 13    Resample filter weights:  $w_{n,j}^{F,\theta} = w_{n,k_j}^{FC,\theta}$  for  $j$  in  $1:J$
  - 14    After-resampling conditional likelihood:  $L_n^{A,\theta,\alpha} = L_n^\phi \frac{\sum_{j=1}^J w_{n,j}^{F,\theta}}{\sum_{j=1}^J w_{n,j}^{P,\theta}}$
  - 15 **end**
- 

Algorithm 1 just repeats MOP- $\alpha$  from the earlier MOP/DOP draft. The final estimate of the likelihood is either based on the after-resampling conditional likelihood estimate

$$L^A(\theta) = \left( \frac{1}{J} \sum_{j=1}^J w_{N,j}^{F,\theta} \right) \prod_{n=1}^N L_n^\phi \quad (1)$$

or the before-resampling estimate,

$$L^B(\theta) = \prod_{n=1}^N \frac{\sum_{j=1}^J w_{N,j}^{P,\theta} g_{N,j}^\theta}{\sum_{j=1}^J w_{N,j}^{P,\theta}}. \quad (2)$$

with  $L^B(\theta)$  presumably having slightly lower variance.

## 2 DOP- $\alpha$

Algorithm 2 is a doubly off policy particle filter, meaning that both `rprocess` and `rmeasure` are computed at  $\phi$  and particles are reweighted to correspond to  $\theta$ . Something like this is needed to deal with discrete latent state spaces, which are beyond the realm of MOP. This algorithm is not plug-and-play. It seems to require transition probabilities (or at least their ratios) between observation times. In the next section, we show how this can be combined with MOP- $\alpha$  to make an algorithm which is practical for over-dispersed continuous time Markov chains

NOTE: THIS IS STILL PROVISIONAL. FOR EXAMPLE, WE PROBABLY NEED dinit SINCE WE CAN'T ASSUME WE CAN DIFFERENTIATE THROUGH TAHT. THE PREVIOUS COP VERSION DID NOT DO THAT. OR, IN PRACTICE, MAYBE ONE CAN DO A DISCRETE OPTIMIZATION OVER THE IVP, IF IT IS ACTUALLY A DISCRETE PARAMETER? ALTERNATIVELY, IT MIGHT BE GOOD TO HAVE A PRIOR OF CONVENIENCE FOR THE IVPS. PERHAPS PUTTIG MASS ON AN INITIAL COMPARTMENT VALUE  $C_0$  GIVEN BY  $\text{round}(\theta_C) - 1, \text{round}(\theta_C), \text{round}(\theta_C) + 1$  where  $\theta_C$  is a continuous parameter for the value of  $C_0$ .

## 3 Process partially off policy (PPOP- $\alpha$ )

OR RENAME TO, E.G., COMPARTMENT MODEL AUTOMATIC DIFFERENTIATION (CMAD)

We suppose that, conditionally on a continuous integrated noise process  $\Sigma(t)$ , the latent process  $X(t)$  is a multinomial Markov compartment model. We can formalize this definition, but it is intended to correspond to the class of models for which an Euler multinomial approximation is appropriate, where the Euler rate between  $t$  and  $t + \delta$  depends on  $\Sigma(t + \delta) - \Sigma(t)$  and  $X(t)$ . We suppose  $K_n$  Euler timesteps between  $t_n$  and  $t_{n+1}$ . In practice, the intermediate times  $\{t_{n_0} = t_n, t_{n,1}, \dots, t_{K_n} = t_{n+1}\}$  may be chosen so that  $\delta_{n,k} = t_{n_{k+1}} - t_{n_k} \leq \delta$  for some  $\delta$ . We write  $X_{n,k} = X(t_{n,k})$  and we note that  $X_n = X_{n,0}$  and  $X_{n+1} = X_{n,K_n}$ .

As for MOP and DOP, PPOP in Algorithm 3 supposes that the algorithm is run first at  $\theta = \phi$  and then at arbitrary  $\theta$ . To differentiate it at  $\theta = \phi$ , we need only one pass, or rather the two passes correspond to the two passes that AD does automatically.

Discounting can be done at each individual Euler timestep,  $t_{n,k}$ , or just at observation times,  $t_n$ . The finer discretization is slightly smoother and respects the mixing that occurs within the interval  $[t_{n-1}, t_n]$ . The extra computation will be negligible in most situations. Therefore, that is what we initially propose.

It remains to define `reulermultinom` and `deulermultinom`. This is carried out separately for departed from each compartment, which are supposed to be infinitesially independent given  $\Sigma(t)$ . Suppose compartments are labeled  $c = 1, \dots, C$ , and we have  $N_{ab}(t)$  counting total transitions from compartment  $a$  to  $b$  by time  $t$ . We formally work with the Markov process  $X(t) = \{N_{ab}(t), (a, b) \in 1 : C \times 1 : C\}$ . together with an initial compartment membership  $X(t_0)$ , with the compartment membership process,  $X_c(t)$ , being defined implicitly.

Then, the Euler multinomial approximation, with rate function  $\rho_{ab}(x, t, \theta)$  and multiplicative integrated

---

**Algorithm 2: Doubly off policy (DOP- $\alpha$ ) SMC:**


---

```

1 Initialize filter particles: simulate  $X_{0,j}^{F,\phi} \sim f_{X_0}(\cdot; \phi)$  for  $j$  in  $1:J$ 
2 Initialize relative weights:  $w_{0,j}^{F,\theta} = 1$  for  $j$  in  $1:J$ 
3 for  $n$  in  $1:N$  do
4   Prediction weights with discounting:  $w_{n,j}^{P,\theta} = (w_{n-1,j}^{F,\theta})^\alpha$  for  $j$  in  $1:J$ 
5   Simulate for prediction:  $X_{n,j}^{P,\phi} \sim f_{X_n|X_{n-1}}(\cdot | X_{n-1,j}^F; \phi)$  for  $j$  in  $1:J$ 
6   Adjust weights:  $w_{n,j}^{P,\theta} = w_{n,j}^{P,\theta} \times \frac{f_{X_n|X_{n-1}}(X_{n,j}^{P,\phi} | X_{n-1,j}^F; \theta)}{f_{X_n|X_{n-1}}(X_{n,j}^{P,\phi} | X_{n-1,j}^F; \phi)}$  for  $j$  in  $1:J$ 
7   Evaluate measurement density:  $g_{n,j}^\theta = f_{Y_n|X_n}(y_n^* | X_{n,j}^{P,\phi}; \theta)$  for  $j$  in  $1:J$ 
8   Before-resampling conditional likelihood:  $L_n^{B,\theta,\alpha} = \frac{\sum_{j=1}^J g_{n,j}^\theta w_{n,j}^{P,\theta}}{\sum_{j=1}^J w_{n,j}^{P,\theta}}$ 
9   Conditional likelihood under  $\phi$ :  $L_n^\phi = \frac{1}{J} \sum_{m=1}^J g_{n,m}^\phi$ 
10  Normalize weights:  $\tilde{g}_{n,j}^\phi = \frac{g_{n,j}^\phi}{J L_n^\phi}$  for  $j$  in  $1:J$ 
11  Apply systematic resampling to select indices  $k_{1:J}$  with  $\text{Prob}(k_j = m) = \tilde{g}_{n,m}^\phi$ 
12  Resample particles:  $X_{n,j}^{F,\phi} = X_{n,k_j}^{P,\phi}$ 
13  Filter weights corrected for resampling:  $w_{n,j}^{FC,\theta} = w_{n,j}^{P,\theta} \times \frac{g_{n,j}^\theta}{g_{n,j}^\phi}$  for  $j$  in  $1:J$ 
14  Resample filter weights:  $w_{n,j}^{F,\theta} = w_{n,k_j}^{FC,\theta}$  for  $j$  in  $1:J$ 
15  After-resampling conditional likelihood:  $L_n^{A,\theta,\alpha} = L_n^\phi \frac{\sum_{j=1}^J w_{n,j}^{F,\theta}}{\sum_{j=1}^J w_{n,j}^{P,\theta}}$ 
16 end

```

---

noise process  $\Sigma(t)$ , with variance depending on  $\theta$ , has the form

$$\{\Delta N_{cb}(t), b \in 1 : C\} \sim \text{multinomial}(X_c, \pi_c)$$

where  $\pi_c = (\pi_{c1}, \dots, \pi_{cC})$  is specified in terms of the individual rates,  $r_{ca} = \rho_{ca} \Delta \Sigma_{ca}$ , and the total rate,  $R_c = \sum_{a \neq c} r_{ca}$  as

$$\pi_{ca} = \begin{cases} (r_{ca}/R_c)(1 - \exp\{-R_c\}) & , \quad a \neq c \\ 1 - \sum_{b \neq c} \pi_{cb} & , \quad a = c \end{cases}$$

The transitions are independent for  $c \in 1 : C$  conditional on  $\Delta \Sigma(t)$  and  $X(t)$ .

---

**Algorithm 3: Process partially off policy (PPOP- $\alpha$ ) SMC:**


---

```

1 Initialize filter particles: simulate  $X_{0,j}^{F,\theta} \sim f_{X_0}(\cdot; \phi)$  for  $j$  in  $1:J$ 
2 Initialize relative weights:  $w_{0,j}^{F,\theta} = \frac{f_{X_0}(X_{0,j}^{F,\theta}; \theta)}{f_{X_0}(X_{0,j}^{F,\theta}; \phi)}$  for  $j$  in  $1:J$ 
3 for  $n$  in  $1:N$  do
4   for  $k$  in  $1:K_n$  do
5     Prediction weights with discounting:  $w_{n,k,j}^{P,\theta} = (w_{n,k-1,j}^{F,\theta})^\alpha$  for  $j$  in  $1:J$ 
6     Simulate  $\Delta\Sigma_k^\theta = \Sigma(t+\delta) - \Sigma(t)$ 
7     Simulate for prediction:  $X_{n,k,j}^{P,\phi} \sim \text{reulermultinom}(X_{n,k-1,j}^F, \rho(\phi, \Delta\Sigma_k^\phi), t_{n,k-1}, t_{n,k})$ 
8     Adjust weights:  $w_{n,k,j}^{P,\theta} = w_{n,j}^{P,\theta} \times \frac{\text{deulermultinom}(X_{n,k-1,j}^F, X_{n,k,j}^F, \rho(\theta, \Delta\Sigma_k^\theta), t_{n,k-1}, t_{n,k})}{\text{deulermultinom}(X_{n,k-1,j}^F, X_{n,k,j}^F, \rho(\phi, \Delta\Sigma_k^\phi), t_{n,k-1}, t_{n,k})}$ 
9   end
10  Evaluate measurement density:  $g_{n,j}^\theta = f_{Y_n|X_n}(y_n^*|X_{n,j}^{P,\phi}; \theta)$  for  $j$  in  $1:J$ 
11  Before-resampling conditional likelihood:  $L_n^{B,\theta,\alpha} = \frac{\sum_{j=1}^J g_{n,j}^\theta w_{n,j}^{P,\theta}}{\sum_{j=1}^J w_{n,j}^{P,\theta}}$ 
12  Conditional likelihood under  $\phi$ :  $L_n^\phi = \frac{1}{J} \sum_{m=1}^J g_{n,m}^\phi$ 
13  Normalize weights:  $\tilde{g}_{n,j}^\phi = \frac{g_{n,j}^\phi}{J L_n^\phi}$  for  $j$  in  $1:J$ 
14  Apply systematic resampling to select indices  $k_{1:J}$  with  $\text{Prob}(k_j = m) = \tilde{g}_{n,m}^\phi$ 
15  Resample particles:  $X_{n,j}^{F,\phi} = X_{n,k_j}^{P,\phi}$ 
16  Filter weights corrected for resampling:  $w_{n,j}^{FC,\theta} = w_{n,j}^{P,\theta} \times \frac{g_{n,j}^\theta}{g_{n,j}^\phi}$  for  $j$  in  $1:J$ 
17  Resample filter weights:  $w_{n,j}^{F,\theta} = w_{n,k_j}^{FC,\theta}$  for  $j$  in  $1:J$ 
18  After-resampling conditional likelihood:  $L_n^{A,\theta,\alpha} = L_n^\phi \frac{\sum_{j=1}^J w_{n,j}^{F,\theta}}{\sum_{j=1}^J w_{n,j}^{P,\theta}}$ 
19 end

```

---