Pseudo-code for simulating the Capture model

Ricardo Carrizo Vergara[1], Marc Kéry[1], Trevor Hefley[2]

[1]Population Biology Group, Swiss Ornithological Institute, 6204 Sempach, Switzerland.

[2]Department of Statistics, Kansas State University, Manhattan, KS 66506, USA.

Reminders: The method here implemented is the *from capture-time to capture-position* method, explained in general terms in Carrizo Vergara et al. (2024, Section 6.2). The underlying free-trajectory model is supposed to depend upon the parameter $\theta_{\tilde{X}}$. The method discretizes the time (regular time grid required). Only one liberation time is considered. This discretization is used to solve numerically the Volterra equation for the density of the capture time

$$f_{T_c}(t) = \alpha \left(\mu_{\tilde{X}(t)|\theta_{\tilde{X}}}(D_c) - \int_{t_0}^{t} \mu_{\tilde{X}(t)|\tilde{X}(u)\in D_c,\theta_{\tilde{X}}}(D_c) f_{T_c}(u) du \right) \tag{1}$$

The `R` function present in the repository is a vectorized implementation of this code, with an extra simplification ad-hoc for the Brownian motion case (see the implementation details Carrizo Vergara et al. (2024, Appendix A)), and can only be applied for capture domains which can be partioned into squares.

Algorithm 1 Ad-hoc simulation method of a discretized version of the Capture model in a time horizon $(t_0, t_H]$ with one liberation time t_L, using a capture-time based simulation.

Input: $N, \theta_{\tilde{X}}, \alpha, t_0, t_L, t_H$, Δt, and a constructive-target-partition of D_c, $A_1, ..., A_m$.

Ensure: $n_L = \frac{t_L - t_0}{\Delta t}$ and $n_{AL} = \frac{t_H - t_L}{\Delta t}$ are integers.

Define $t_1, ..., t_{n_L+n_{AL}}$ a regular discretization time grid with $t_k = t_{k-1} + \Delta t$ for $k = 1, ..., n_L + n_{AL}$.

Define $PP1S$ a matrix of dimensions $(n_L + n_{AL}) \times m$ such that $PP1S_{k,l} \leftarrow \mu_{\tilde{X}(t_k)|\theta_{\tilde{X}}}(A_l)$.

Define $PP1$ a vector of dimension $n_L + n_{AL}$ such that $PP1_k \leftarrow \mu_{\tilde{X}(t_k)|\theta_X}(D_c) = \sum_{l=1}^{m} \mu_{\tilde{X}(t_k)|\theta_{\tilde{X}}}(A_l)$.

Verify $0 < PP1_k \leqslant 1$ for all $k = 1, ..., n_L + n_{AL}$.

Define $PP2S$ a 4-dimensional array with dimensions $(n_L + n_{AL}) \times (n_L + n_{AL}) \times m \times m$, such that $PP2S_{k,k',l,l'} \leftarrow \mu_{\tilde{X}(t_k),\tilde{X}(t_{k'})|\theta_{\tilde{X}}}(A_l \times A_{l'})$. It is enough to define it for $k' \leqslant k$.

Define $PP2$ a matrix of dimensions $(n_L + n_{AL}) \times (n_L + n_{AL})$ such that $PP2_{k,k'} \leftarrow \mu_{\tilde{X}(t_k),\tilde{X}(t_{k'})|\theta_{\tilde{X}}}(D_c \times D_c) = \sum_{l=1}^{m} \sum_{l'=1}^{m} \mu_{\tilde{X}(t_k),\tilde{X}(t_{k'})|\theta_{\tilde{X}}}(A_l \times A_{l'})$. It is enough to define it for $k' \leqslant k$.

Verify $0 \leqslant PP2_{k,k'} \leqslant 1$ for every k, k', and $\max\limits_{k'=1,...,n_L+n_{AL}} PP2_{k,k'} \leqslant PP1_k$ for every k.

Define CPP a lower-triangular matrix of dimensions $(n_L + n_{AL}) \times (n_L + n_{AL})$ such that $CPP_{k,k'} \leftarrow \frac{PP2_{k,k'}}{PP1_{k'}}$ for $k' \leqslant k$.

Define the vector $\vec{f} = (f_1, ..., f_{n_L+n_{AL}})$ as the solution of the triangular linear system obtained from the discretization of the Volterra equation (1):

$$\left(I + \alpha \Delta t CPP\right) \vec{f} = \alpha PP1, \tag{2}$$

where I is the identity matrix of dimension $n_L + n_{AL}$.

Define the vector of probabilities $PTc1 = (\Delta t f_1, ..., \Delta t f_{n_L+n_{AL}}, 1 - \Delta t \sum_{k=1}^{n_L+n_{AL}} f_k)$.

Define $\vec{\phi}$ a vector of dimension $n_L + n_{AL}$ given by $\phi_k = \frac{f_k}{PP1_k}$.

Define $FT2$ a matrix of dimensions $n_{AL} \times n_L$ given by $FT2_{k,k'} \leftarrow \alpha CPP_{k'+k,k'} \frac{\phi_{k'+k}}{\phi_{k'}}$.

Define $\boldsymbol{Q} = (Q_{k,l})_{k,l}$ as a zero matrix of dimensions $(n_L + n_{AL}) \times m$.

for $j = 1, ..., N$ **do**

Simulate the first-capture-time index of individual j, $k_{c1}^{(1)}$, according to

$$k_{c1}^{(j)} = \begin{cases} k & \text{with probability } PTc1_k, \text{ for } k = 1, ..., n_L + n_{AL} \\ \infty & \text{with probability } PTc1_{n_L+n_{AL}+1}. \end{cases} \tag{3}$$

Define both the first-time-position index $l_{c1}^{(j)}$ and the second-time-position index $l_{c2}^{(j)}$ of individual j as `NULL`.

if $k_{c1}^{(j)} \leqslant n_L$ **then**

Define the vector $PTc2^{(j)} = (\Delta t FT2_{1,k_{c1}^{(j)}}, ..., \Delta t FT2_{n_{AL},k_{c1}^{(j)}}, 1 - \Delta t \sum_{k=1}^{n_{AL}} FT2_{k,k_{c1}^{(j)}})$.

Simulate the speculative second-capture-time-lag index of individual j, $\tilde{k}_{c2}^{(j)}$, according to

$$\tilde{k}_{c2}^{(j)} = \begin{cases} \tilde{k} & \text{with probability } PTc2_{\tilde{k}}^{(j)} \text{ for } \tilde{k} = 1, ..., n_{AL}, \\ \infty & \text{with probability } PTc2_{n_{AL}+1}^{(j)}. \end{cases} \tag{4}$$

if $\tilde{k}_{c2}^{(j)} \leqslant n_{AL}$ **then**
Simulate $l_{c1}^{(j)}, l_{c2}^{(j)}$ according to

$$(l_{c1}^{(j)}, l_{c2}^{(j)}) = (l', l) \text{ with probability } \frac{PP2S_{k_{c1}^{(j)}+\tilde{k}_{c2}^{(j)}, k_{c1}^{(j)}, l, l'}}{PP2_{k_{c1}^{(j)}+\tilde{k}_{c2}^{(j)}, k_{c1}^{(j)}}}, \text{ for } l, l' = 1, ..., m. \tag{5}$$

$Q_{k_{c1}^{(j)}, l_{c1}^{(j)}} \leftarrow Q_{k_{c1}^{(j)}, l_{c1}^{(j)}} + 1.$
$Q_{n_L+\tilde{k}_{c2}^{(j)}, l_{c2}^{(j)}} \leftarrow Q_{n_L+\tilde{k}_{c2}^{(j)}, l_{c2}^{(j)}} + 1.$
else
Simulate $l_{c1}^{(j)}$ according to

$$l_{c1}^{(j)} = l \text{ with probability } \frac{PP1S_{k_{c1}^{(j)}, l} - \frac{\alpha \Delta t}{\phi_{k_{c1}^{(j)}}} \sum_{\tilde{k}=1}^{n_{AL}} \sum_{l'=1}^{m} PP2S_{k_{c1}^{(j)}+\tilde{k}, k_{c1}^{(j)}, l', l} \phi_{k_{c1}^{(j)}+\tilde{k}}}{PP1_{k_{c1}^{(j)}} PTc2_{n_{AL}+1}^{(j)}}, \text{ for } l = 1, ..., m. \tag{6}$$

$Q_{k_{c1}^{(j)}, l_{c1}^{(j)}} \leftarrow Q_{k_{c1}^{(j)}, l_{c1}^{(j)}} + 1.$
end if
else if $n_L < k_{c1}^{(j)} \leqslant n_L + n_{AL}$ **then**
Simulate $l_{c1}^{(j)}$ according to

$$l_{c1}^{(j)} = l \text{ with probability } \frac{PP1S_{k_{c1}^{(j)}, l}}{PP1_{k_{c1}^{(j)}}}, \text{ for } l = 1, ..., m. \tag{7}$$

$Q_{k_{c1}^{(j)}, l_{c1}^{(j)}} \leftarrow Q_{k_{c1}^{(j)}, l_{c1}^{(j)}} + 1.$
end if
end for
for $l = 1, ..., m$ **do**
for $k = 1, ..., n_L + n_{AL}$ **do**
if $k \neq 1$ and $k \neq n_L + 1$ **then**
$Q_{k,l} \leftarrow Q_{k,l} + Q_{k-1,l}$
end if
end for
end for
return $\boldsymbol{Q}$

References

Carrizo Vergara, R., Kéry, M., & Hefley, T. (2024). Movement-based models for abundance data. *arXiv preprint arXiv:2407.13384*.