Noname manuscript No.
(will be inserted by the editor)

Supplementary Material for Determining Parameter Redundancy of Multi-state Mark-recapture Models for Sea Birds

Diana J. Cole

This is the supplementary material for the paper Determining Parameter Redundancy of Multi-state Mark-recapture Models for Sea Birds.

Appendix 1: Proof of Simpler Exhaustive Summary

The Maple code form this proof can be found in Maple worksheet `simexsumproof.mw`. To show that Table 1, from the main paper, is an exhaustive summary, first consider the case of no unobservable states. Starting from $N = 3$, $S = 2$ we show that for transition and recapture matrices

$$\boldsymbol{\Phi}_t = \begin{bmatrix} a_{1,1}(t) & a_{1,2}(t) \\ a_{2,1}(t) & a_{2,2}(t) \end{bmatrix} \; \mathbf{P}_t = \begin{bmatrix} p_1(t) & 0 \\ 0 & p_2(t) \end{bmatrix}$$

a possible reparameterisation with 10 elements is

$$\mathbf{s} = [p_1(2)a_{1,1}(1), p_2(2)a_{2,1}(1), p_1(2)a_{1,2}(1), p_2(2)a_{2,2}(1), p_1(2), p_2(2), p_1(3)a_{1,1}(2), ..., p_2(3)a_{2,2}(2)]^T .$$

We rewrite the original exhaustive summary, consisting of the non-zero p-array terms, in terms of s_i. This is then differentiated with respect to the s_i to form the derivative matrix $\mathbf{D}_s$. The derivative matrix has full rank 10, so the reparameterisation $\mathbf{s}$ is an exhaustive summary. Now extend this to $N = 4$. The extra reparameterisation terms are

$$\mathbf{s}^{'} = [p_1(4)a_{1,1}(3), ..., p_2(4)a_{2,2}(2), p_1(3), p_2(3)]^T .$$

D. Cole
School of Mathematics, Statistics and Actuarial Science University of Kent, Canterbury, Kent, CT2 7NF, UK.
Tel: +44-(0)1227-823664
E-mail: D.J.Cole@kent.ac.uk

Extra original exhaustive summary terms are also added, which are differentiated with respect to $\mathbf{s}^{'}$ to form the derivative matrix $\mathbf{D}_{ex}$. The rank of $\mathbf{D}_{ex}$ is of full rank 6, therefore by the extension theorem and by induction (Cole *et al*, 20010)

$$\mathbf{s} = [p_1(2)a_{1,1}(1), ..., p_2(2)a_{2,2}(1), p_1(2), p_2(2), ..., p_1(N-1), p_2(N-1), p_1(N)a_{1,1}(N-1), ..., p_N(N)a_{2,2}(N-1),]^T$$

will always be an exhaustive summary for any $N \geq 3$.

Next consider extending to $S = 3$ again starting from $N = 3$. The transition and recapture matrices are

$$\boldsymbol{\Phi}_t = \begin{bmatrix} a_{1,1}(t) & a_{1,2}(t) & a_{1,3}(t) \\ a_{2,1}(t) & a_{2,2}(t) & a_{2,3}(t) \\ a_{3,1}(t) & a_{3,2}(t) & a_{3,3}(t) \end{bmatrix}$$
$$\mathbf{P}_t = \begin{bmatrix} p_1(t) & 0 & 0 \\ 0 & p_2(t) & 0 \\ 0 & 0 & p_3(t) \end{bmatrix} . \quad (1)$$

The reparameterisation has the extra terms

- $p_3(t + 1)a_{3,j}(t)$ for $j = 1, 2, 3$ and $t = 1, 2$,
- $p_i(t + 1)a_{i,3}(t)$ for $i = 1, 2$ and $t = 1, 2$,
- $p_3(2)$,

and in a similar way we use the extension theorem and induction to deduce that when all states are observable an exhaustive summary consists of the terms:

- $p_i(t + 1)a_{i,j}(t)$ for $t = 1, ..., N - 1$, $i = 1, .., S$ and $j = 1, .., S$,
- $p_i(t)$ for $t = 2, ..., N - 1$ and $i = 1, .., S$,

which are the only terms of the exhaustive summary given by Table 1, that apply when $U = 0$.

Then we move on to consider unobservable states. The first case we consider is $S = 3$, $N = 4$, with one unobservable state, so that $U = 1$. The transition and recapture matrices are the same as in equation 1, with

$p_3(t) = 0$. A possible reparameterisation of length 23 has elements of the form given in Table 1. We rewrite the original exhaustive summary, consisting of the non-zero p-array terms, in terms of s_i. This is then differentiated with respect to the s_i to form the derivative matrix $\mathbf{D}_s$. The derivative matrix has full rank 23, so the reparameterisation given by 1 is an exhaustive summary, for $S = 3$, $N = 4$ and $U = 1$. This can then be extend to $N = 4$. There are 10 extra reparameterisation terms. The extra original exhaustive summary terms are analysed with respect to these 10 extra reparameterisation terms and the exhaustive summary is applied in three parts; the first two are shown to be full rank in the Maple worksheet. The third part contains no extra reparameterisation parameters. Therefore by the extension theorem and by induction Table 1 is an exhaustive summary, for $S = 3$, $N \geq 4$ and $U = 1$.

Next we consider extending the case $S = 3$, $N = 4$, $U = 1$ to the case $S = 4$, $N = 4$, $U = 1$. For convenience of the proof, we continue to label the 3rd state as unobservable and introduce a fourth new observable state. (This is contrast to the exhaustive summary presented in Table 1, where the fourth state would be labeled as unobservable.) The transition and recapture matrices are

$$\boldsymbol{\Phi}_t = \begin{bmatrix} a_{1,1}(t) & a_{1,2}(t) & a_{1,3}(t) & a_{1,4}(t) \\ a_{2,1}(t) & a_{2,2}(t) & a_{2,3}(t) & a_{2,4}(t) \\ a_{3,1}(t) & a_{3,2}(t) & a_{3,3}(t) & a_{3,4}(t) \\ a_{4,1}(t) & a_{4,2}(t) & a_{4,3}(t) & a_{4,4}(t) \end{bmatrix}$$

$$\mathbf{P}_t = \begin{bmatrix} p_1(t) & 0 & 0 & 0 \\ 0 & p_2(t) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_4(t) \end{bmatrix}.$$

By examining the extra original exhaustive summary terms with respect to the extra reparameterisation terms that are added:

- $p_4(t+1)a_{4,j}(t)$ for $j = 1, 2, 4$ and $t = 1, ..., 4$,
- $p_i(t+1)a_{i,4}(t)$ for $i = 1, 2$ and $t = 1, ..., 4$,
- $p_4(t)$ for $t = 2, ..., 4$,
- $p_4(t+1)a_{4,3}(t)a_{3,1}(t-1)$ for $t = 2, ..., 4$,
- $\dfrac{a_{3,4}(t-1)}{a_{3,1}(t-1)}$ for $t = 2, ..., 4$,

we show in the Maple worksheet using the extension theorem and by induction, that once the states are relabeled, that exhaustive summary presented in Table 1 is an exhaustive summary for $S \geq 3$, $N \geq 4$ and $U = 1$.

Finally we consider extending the case $S = 3$, $N = 7$, $U = 1$ to the case $S = 4$, $N = 7$, $U = 2$. From the results above above we already know that Table 1 is an exhaustive summary for $S = 3$, $N = 7$, $U = 1$. We then use the extension theorem once more. Examining the extra original exhaustive summary terms with respect to the extra reparameterisation terms that are added:

- $p_4(t+1)a_{i,4}(t)a_{4,1}(t-1)$ for $t = 2, ..., 6$ and $i = 1, 2$,
- $\dfrac{a_{4,2}(t-1)}{a_{4,1}(t-1)}$ for $t = 2, ..., 6$,
- $\dfrac{a_{4,3}(t-1)a_{3,1}(t-2)}{a_{4,1}(t-1)}$ for $t = 3, ..., 6$,
- $\dfrac{a_{3,4}(t-1)a_{4,1}(t-2)}{a_{3,1}(t-1)}$ for $t = 3, ..., 6$,
- $\dfrac{a_{4,4}(t-1)a_{4,1}(t-2)}{a_{4,1}(t-1)}$ for $t = 3, ..., 6$,

we show in the Maple worksheet using the extension theorem and by induction, that the exhaustive summary presented in Table 1 is an exhaustive summary for $S \geq 4$, $N \geq 7$ and $U = 2$.

We have proved the case $S \geq 3$, $N \geq 4$ and $U = 1$. Then consider any other S and U such that $S - U > 1$. There will always be a $S' = S - 1$ and a $U' = U - 1$ so that a similar move as from $S = 3$, $U = 1$ to $S = 4$, $U = 2$ can be applied. All that is required is to start at an N large enough so that there will be more original exhaustive summary terms than reparameterisation terms. There are $(S^2 + S - 2U)(N - 1) - S - 2U(S - 1)$ terms in the reparameterisation and $\frac{1}{2}(N^2 - N)(S - U)^2$ terms in the original exhaustive summary. So this results in a restriction on N of $(S^2 + S - 2U)(N - 1) - S - 2U(S - 1) < \frac{1}{2}(N^2 - N)(S - U)^2$. Therefore by induction the exhaustive summary presented in Table 1 is an exhaustive summary for all cases as long as $U - S > 1$ and $(S^2 + S - 2U)(N - 1) - S - 2U(S - 1) < \frac{1}{2}(N^2 - N)(S - U)^2$.

Appendix 2: Proof of Recruitment Simpler Exhaustive Summary

This section provides the proof for the recruitment exhaustive summary given in Table 4, or the main paper. The Maple code for this proof can be found in Maple worksheet `recruitmentproof.mw`.

The proof starts by considering the case of $N = 3$ years of ringing and recovery, $k = 3$ years of recruitment and $y = 2$ recruiting classes. It is shown in the `recruitmentproof.mw` that

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} p_2\sigma_{5,1} \\ p_2 \\ p_3\sigma_{5,2} \end{bmatrix},$$

is an exhaustive summary. This is shown by rewriting the original exhaustive summary, consisting of the entries of the p-array, in terms of $\mathbf{s}$ and then finding the derivative matrix of this with respect to the s_i. This derivative matrix has full rank 3, so $\mathbf{s}$ is an exhaustive summary. As well as $N = 3$ we then consider each of $N = 4$,$N = 5$ and $N = 6$ to show that

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \\ s_{11} \\ s_{12} \\ s_{13} \\ s_{14} \end{bmatrix} = \begin{bmatrix} \left.\begin{matrix} p_2\sigma_{5,1} \\ p_2 \\ p_3\sigma_{5,2} \end{matrix}\right\} N \geq 3 \\ \left.\begin{matrix} p_3 \\ p_4\sigma_{5,3} \\ p_4\sigma_{3,3}\alpha_{3,3}\sigma_{2,2}\sigma_{1,1} \end{matrix}\right\} N \geq 4 \\ \left.\begin{matrix} p_4 \\ p_5\sigma_{5,4} \\ p_5\sigma_{3,4}\alpha_{3,4}\sigma_{2,3}\sigma_{1,2} \\ p_5\sigma_{4,4}\alpha_{4,4}(1-\alpha_{3,3})/\alpha_{3,3} \end{matrix}\right\} N \geq 5 \\ \left.\begin{matrix} p_5 \\ p_6\sigma_{5,5} \\ p_6\sigma_{3,5}\alpha_{3,5}\sigma_{2,4}\sigma_{1,3} \\ p_6\sigma_{4,5}\alpha_{4,5}(1-\alpha_{3,4})/\alpha_{3,4} \end{matrix}\right\} N \geq 6 \end{bmatrix},$$

is an exhaustive summary by considering ranks of derivative matrices. The extension theorem (Catchpole and Morgan, 1997 and Cole *et al*, 2010) is also applied to show that each additional year of ringing and recovery would add the exhaustive summary terms:

$$\begin{bmatrix} p_{N-1} \\ p_N\sigma_{5,N-1} \\ p_N\sigma_{3,N-1}\alpha_{3,N-1}\sigma_{2,N-2}\sigma_{1,N-3} \\ p_N\sigma_{4,N-1}\alpha_{4,N-1}(1-\alpha_{3,N-2})/\alpha_{3,N-2} \end{bmatrix}.$$

Next consider increasing k by 1. It is shown in the Maple worksheet that apart from changing numbering, the only difference is that the exhaustive summary term

$$p_N\sigma_{3,N-1}\alpha_{3,N-1}\sigma_{2,N-2}\sigma_{1,N-3}$$

is replaced by

$$p_N\sigma_{4,N-1}\alpha_{4,N-1}\sigma_{3,N-2}\sigma_{2,N-3}\sigma_{1,N-4}$$

and it and the subsequent term each appear when N is one greater. This pattern continues as k is increased further.

Now consider again the case where $k = 3$ and instead increase y by 1 so $y = 3$. In each case of $N = 3, \ldots, 6$ the extension theorem can be used to show that adding an extra y adds an extra exhaustive summary term $s_{14} = p_N\sigma_{5,N-1}\alpha_{5,N-1}(1-\alpha_{4,N-2})/\alpha_{4,N-2}$ when $N \geq 6$, but that $\mathbf{s}$ is still an exhaustive summary. Hence by induction we can prove that an exhaustive summary for the recruitment model is given by Table 4.

Appendix 3: 9-state recruitment example

The 9-state recruitment model of Hunter and Caswell (2009) has $k = 4$ and $y = 5$. The Maple procedure `recruitment` finds the exhaustive summary given in Table 4. Using Maple code of the form:

```
> N := 5:
> kappa := recruitment(N, 4, 5):
> pars := <seq(op(i,indets(kappa)),
  i=1..nops(indets(kappa)))>:
> DD := Dmat(kappa, pars):
> r := Rank(DD); d := Dimension(pars)-r;
  pp := Dimension(pars);
> Estpars(DD, pars);
```

it is possible to find the rank and deficiency and estimable parameter combinations for each of $N \geq 3$. These results are displayed in Table 7. It is possible to find the general case for any $N \geq 8$, as the exhaustive summary is in reduced-form, and therefore the exhaustive summary is a vector of the minimum length. The minimum length of an exhaustive summary is equal to the rank of the derivative matrix, therefore the rank is $7N - 33$ for any $N \geq 8$. There are 15 new parameters each time (apart from first few) so the number of parameters is $15N - 74$. The deficiency is then calculated from $d = p - r$.

Appendix 4: Maple Procedures

This appendix consists of Maple procedures for using the derivative matrix method to investigate parameter redundancy in multi-state mark-recapture models.

The procedure `Dmat` finds the derivative matrix given a vector of exhaustive summary term (`kappa`) and a vector of parameters (`pars`).

```
Dmat:=proc(kappa,pars)
local DD1, i, j;
description "Form the derivative matrix";
with(LinearAlgebra);
DD1:=Matrix(1..Dimension(pars),1..Dimension
     (kappa));
for i to Dimension(pars) do
    for j to Dimension(kappa) do
        DD1[i,j]:=diff(kappa[j],pars[i])
    end do
end do;
DD1
end proc
```

The procedure `Estpars` finds the estimable set of parameters given a derivative matrix (`DD1`) and a vector of parameters (`pars`).

N	r	d	p	Estimable Parameters
3	3	1	4	$\sigma_{9,1}, p_2, p_3\sigma_{9,2}$
4	5	1	6	$\sigma_{9,1}, \sigma_{9,2}, p_2, p_3, p_4\sigma_{9,3}$
5	8	5	13	$\sigma_{9,1}, \ldots, \sigma_{9,3}, p_2 \ldots, p_4, p_5\sigma_{9,4}, p_5\sigma_{4,4}\alpha_{4,4}\prod_{j=1}^{3}\sigma_{j,j}$
6	12	10	22	$\sigma_{9,1}, \ldots, \sigma_{9,4}, p_2 \ldots, p_5, p_6\sigma_{9,5}, \sigma_{4,4}\alpha_{4,4}\prod_{j=1}^{3}\sigma_{j,j},$ $p_6\sigma_{4,5}\alpha_{4,5}\prod_{j=1}^{3}\sigma_{j,j+1}, \frac{p_6\alpha_{5,5}\sigma_{5,5}(1-\alpha_{4,4})}{\alpha_{4,4}}$
7	17	16	33	$\sigma_{9,1}, \ldots, \sigma_{9,5}, p_2 \ldots, p_6, p_7\sigma_{9,6}, \sigma_{4,4}\alpha_{4,4}\prod_{j=1}^{3}\sigma_{j,j},$ $\sigma_{4,5}\alpha_{4,5}\prod_{j=1}^{3}\sigma_{j,j+1}, \frac{\alpha_{5,5}\sigma_{5,5}(1-\alpha_{4,4})}{\alpha_{4,4}}, p_7\sigma_{4,6}\alpha_{4,6}\prod_{j=1}^{3}\sigma_{j,j+2},$ $\frac{p_7\alpha_{5,6}\sigma_{5,6}(1-\alpha_{4,5})}{\alpha_{4,5}}, \frac{p_7\alpha_{6,6}\sigma_{6,6}(1-\alpha_{5,5})}{\alpha_{5,5}}$
N	$7N-33$	$8N-41$	$15N-74$	$N \geq 8$

Table 7 Parameter Redundancy in the 9-state Recruitment Model.

```
Estpars:=proc(DD1,pars)
local r, d, alphapre, alpha, PDE, FF, i, ans;
description "Finds the estimable set of
             parameters";
with(LinearAlgebra);
r := Rank(DD1);
d := Dimension(pars)-r;
alphapre:=NullSpace(Transpose(DD1));
alpha:=Matrix(d, Dimension(pars));
PDE:=Vector(d);
FF:=f(seq(pars[i],i=1..Dimension(pars)));
for i to d do
    alpha[i,1..Dimension(pars)]:=alphapre[i];
    PDE[i]:=add((diff(FF,pars[j]))*
           alpha[i, j],j=1..Dimension(pars)):
end do;
ans := pdsolve({seq(PDE[i] = 0, i = 1 .. d)})
end proc
```

The procedure `simexsum` finds the simple exhaustive summary for a multi-state model. The inputs are a transition matrix `A`, a recapture matrix `P` and the number of years of ringing and recovery N. See the description for details and restrictions.

The procedure `recruitment` finds the simple exhaustive summary for the recruitment model. The inputs are N, the number of years of ringing and recovery, k, the age at recruitment, y the number of recruiting classes. Note that $k \geq 3$, and $y \geq 2$.

```
simexsum:=proc (A, P, N)
local S, U, i, j, kappa, kappaindex, tt, k, test;
description "Finds a simpler exhaustive summary. A is the transition matrix and must be square. If A is time
dependent the letter t must be used in a subscript to represent time. P is a diagonal recovery matrix and must
be a square matrix with entries only on the diagonal. Unobservable states must be numbered as the last states, and
must have a zero in the appropriate diagonal entry. N is the number of years of the study, with N-1 years of marking
and N-1 years of recovery. If kappa is returned as zero, this means the general exhaustive summary is not valid for
that N, try a greater N";
with(LinearAlgebra);
S := Dimension(A)[1]; U := 0;
for i from S by -1 to 1 do
    if P[i, i] = 0 then U:=U+1: end if
end do;
if (S^2+S-2*U)*(N-1)-S-2*U*(S-1) < ((1/2)*N^2-(1/2)*N)*(S-U)^2 and 1 < S-U then
    kappa := Vector((S^2+S-2*U)*(N-1)-S-2*U*(S-1));
    kappaindex := 1;
    for tt to N-1 do
        for i to S-U do
            for j to S-U do
                kappa[kappaindex]:=(eval(P[i,i],t=tt+1))*(eval(A[i,j],t=tt)); kappaindex:=kappaindex+1
            end do
        end do
    end do;
    for tt from 2 to N-1 do
        for i to S-U do
            kappa[kappaindex] := eval(P[i, i], t = tt);
            kappaindex := kappaindex+1
        end do
    end do;
    for tt from 2 to N-1 do
        for i to S do
            for j from S-U+1 to S do
                if A[j, 1] <> 0 then
                    if i <= S-U then
                        kappa[kappaindex]:=(eval(P[i, i],t=tt+1))*(eval(A[i,j],t=tt))*(eval(A[j,1],t=tt-1));
                        kappaindex:=kappaindex+1
                    end if;
                    if 2 <= i and i <= S-U then
                        kappa[kappaindex]:=(eval(A[j,i],t=tt-1))/(eval(A[j,1],t=tt-1));
                        kappaindex:=kappaindex+1
                    end if;
                    if S-U < i and 2 < tt then
                        kappa[kappaindex]:=(eval(A[j i],t=tt-1))*
                         (eval(A[j,1],t=tt-2))/(eval(A[j, 1],t = tt-1));
                        kappaindex:=kappaindex+1
                    end if
                else
                    test := 0;
                    for k from 2 to S while test = 0 do
                        if A[j, k] <> 0 then
                            if i <= S-U then
                                kappa[kappaindex]:=(eval(P[i,i],t=tt+1))*(eval(A[i,j],t = tt))*(eval(A[j,k],t=tt-1));
                                kappaindex := kappaindex+1
                             end if;
                             if 2 <= i and i <= S-U then
                                kappa[kappaindex]:=(eval(A[j,i],t=tt-1))/(eval(A[j,k],t = tt-1));
                                kappaindex:=kappaindex+1
                             end if;
                             if S-U < i and 2 < tt then
                                kappa[kappaindex]:=(eval(A[j,i],t=tt-1))*(eval(A[j,k],t=tt-2))/(eval(A[j,k],t=tt-1));
                                kappaindex:=kappaindex+1
                             end if;
                             test:=1
                        end if: end do: end if:
                end do:  end do: end do:
else
    kappa := 0
end if;
kappa
end proc
```

```
recruitment := proc (N, k, y)
local i, t, sizekappa, kappa, indexkappa;
description "Gives an exhaustive summary for the recruitment model with N-1 years
of ringing N-1 years of recovery, recruitment age k, recruitment classes y";
with(LinearAlgebra);
sizekappa := 0;
for t to N-1 do
    sizekappa:=sizekappa+1;
    if 2 <= t then
        sizekappa:=sizekappa+1
    end if;
    if k <= t then
        sizekappa:=sizekappa+1
    end if
end do;
for i from k+1 to y+k-1 do
    for t from i to N-1 do
        sizekappa:=sizekappa+1
    end do
end do;
kappa := Vector(sizekappa);
indexkappa:=1;
for t to N-1 do
    kappa[indexkappa]:=p[k+y,t+1]*sigma[k+y,t];
    indexkappa := indexkappa+1;
    if 2 <= t then
        kappa[indexkappa]:=p[k+y, t];
        indexkappa:=indexkappa+1
    end if;
    if k <= t then
        kappa[indexkappa]:=p[k+y,t+1]*sigma[k,]*alpha[k,t]*(product(sigma[j,t+j-k],j=1..k-1));
        indexkappa:=indexkappa+1
    end if
end do;
for i from k+1 to y+k-1 do
    for t from i to N-1 do
        kappa[indexkappa]:=p[k+y,t+1]*sigma[i,t]*alpha[i,t]*(1-alpha[i-1,t-1])/alpha[i-1,t-1];
        indexkappa:=indexkappa+1
    end do
end do;
kappa
end proc
```