

AD Model Builder

(Comparison of estimators for mark-recapture models: random effects, hierarchical Bayes, and AD Model Builder)

Mark N. Maunder, Hans J. Skaug,
and Simon D. Hoyle.

Run models

AD Model Builder nearly exclusively
used in fisheries



"We need easy ways to embed general and flexible random effects into extant capture- recapture models, without each time deriving estimators."

(Burnham and White 2002).

Outline

- Random effects models
- Basics of AD Model Builder
- AD Model Builder Mark-recapture example
- Three applications
 - European dipper
 - Yellow-eyed penguin
 - Blackfooted Albatross

Check MCMC

Random effects likelihood: hierarchical

$$L(\theta) = \int f(y, \theta, v | u) h(u) du$$

Parameter of interest Other nuisance parameters
How to Eliminate? Random effect
Integrate

The diagram illustrates the hierarchical nature of the random effects likelihood. At the top right, a blue arrow points from the text "Random effect distribution" to the term "h(u)" in the equation. Below the equation, three red arrows point upwards from the text labels "Parameter of interest", "Other nuisance parameters", and "Random effect Integrate" to their respective components in the likelihood function: "f(y, theta, v | u)", "h(u)", and "du".

Inference frame works

- Frequentist
 - Simultaneous maximization (nuisance parameters) and integration (random effects)
- Bayesian
 - Integration for all parameters
 - Priors for all parameters

AD Model Builder (ADMB)

- A programming environment for estimating the parameters of parameter rich nonlinear models

Automatic Differentiation (AD): Reduces run time and improves stability

- Machine precision derivatives for optimization algorithm
- Precompiled derivative code for common functions (e.g. matrix algebra)
- Chain rule for other calculations
- Requires less than 5 times the original calculations to provide the derivative values (Griewank 2000; Skaug and Fournier 2006).
- Numerical derivatives requires the number of parameters + 1 times the time to conduct the original calculations
- Inaccuracy of the numerical derivative approximations causes instability in the optimization process and produces unreliable results for ill-conditioned problems.

Estimation methods

- MCMC routine to perform Bayesian integration
 - Uses the mode of the posterior, estimated by the optimization routine, to initiate the MCMC chain
 - Uses the covariance matrix to develop the jumping rule
- Laplace approximation to implement random effects
 - Uses automatic second derivatives
 - Can be used to develop the sampling distribution for use in an importance sampling method to integrate out the random effect parameters.
- Automatic profile likelihood calculations

Other features

- Bounds
- Phases
- Standard deviations and correlation matrix
- Parallel processing
- Many other features

Files

- Input

.tpl	Code file
.dat	Data file
.pin	Initial parameter values file
- Output

.par	Parameter estimate file
.std	Standard deviation file
.cor	Correlation matrix file
.rep	User defined output file

Mechanics

- Most of the code used to developed the model is C/C++ or overloaded C++ functions and operators.
- Key words used in ADMBs template system and ADMB functions
- DATA_SECTION
 - Definitions of data to use in the model
 - Variables that are used for intermediate calculations that don't require derivative calculations (e.g. used for data manipulation).
- PARAMETER_SECTION
 - Definitions of the parameters of the model to be estimated
 - Variables that are used for intermediate calculations that do require derivative calculations (i.e. are functions of the model parameters).
- PROCEDURE_SECTION
 - Code is written to develop the model (i.e. the calculation of the objective function from the data and the model parameters).
- REPORT_SECTION
 - User defined output
- *.dat file contains all the data that was defined in the DATA_SECTION.
- *.pin file provides initial values for all the estimable model parameters defined in the PARAMETER_SECTION.

Simple Least Squares Regression (Simple.TPL)

```

DATA_SECTION
init_int nobs
init_vector y(1,nobs)
init_vector x(1,nobs)
PARAMETER_SECTION
init_number a
init_number b
objective_function_value f
PROCEDURE_SECTION
for (int i=1;i<=nobs;i++)
{
  f+=pow(y(i)-(a+b*x(i)),2);
}

```

$$SS = \sum_i (y_i - [a + bx_i])^2$$

Mark-recapture model

- Call the model MR.tpl
- Survival and probability of recapture unrestricted time-effects
- Multinomial likelihood
- Minimizer: uses negative log-likelihood
- Use European dipper example from Lebreton et al. (1992)

DATA_SECTION

```
DATA_SECTION
init_int NRCperiods
init_vector Releases(1,NRCperiods)
ivector temp(1,NRCperiods)
!temp.fill_seqadd(1,1);
init_matrix data(1,NRCperiods,temp,NRCperiods)
```

MR.dat file

```
# init_int NRCperiods
6
#init_vector Releases(1,NRCperiods)
22 60 78 80 88 98
#init_matrix data(1,NRCperiods,temp,NRCperiods)
11 2 0 0 0 0
24 1 0 0 0
34 2 0 0
45 1 2
51 0
52
```

Parameter definition

```
init_bounded_vector S(Lower index, upper index, LB, UB, phase)
```

PARAMETER_SECTION

```
PARAMETER_SECTION
init_bounded_vector estS(1,NRCperiods
-1,0,1,1)
init_bounded_vector p(1,NRCperiods,0,1,2)
vector S(1,NRCperiods)
matrix Scum(1,NRCperiods,1,NRCperiods)
matrix Pcum(1,NRCperiods,1,NRCperiods)
objective_function_value f
```

MR.pin

```
#init_bounded_vector estS(1,NRCperiods-1,0,1,1)
0.7 0.7 0.7 0.7 0.7
#init_bounded_vector p(1,NRCperiods,0,1,2)
0.7 0.7 0.7 0.7 0.7 0.7
```

PROCEDURE_SECTION

```
PROCEDURE_SECTION
S(1,NRCperiods-1)=estS;
S(NRCperiods)=1; //so p is combined S and p for last year
for (int i=1;i<=NRCperiods;i++)
{
    Scum(i,i)=S(i);
    Pcum(i,i)=1;
    for (int j=i+1;j<=NRCperiods;j++)
    {
        Scum(i,j)=Scum(i,j-1)*S(j);
        Pcum(i,j)=Pcum(i,j-1)*(1.0-p(j-1));
    }
    Pcum(i)=elem_prod(Pcum(i),p);
    for (j=i;j<=NRCperiods;j++)
        f+=-data(i,j)*log(Scum(i,j)*Pcum(i,j));
    f+=-(Releases(i)-sum(data(i)))*log(1-
    sum(elem_prod(Scum(i),Pcum(i))));
}
```

REPORT_SECTION

```
REPORT_SECTION
report<<"S "<<S<<endl;
report<<"p "<<p<<endl;
```

Running the model

- The model is compiled using the command
admb MR
- The model is run using the command
MR

Creating Executable

- Translate template into C++ code (tpl2cpp.exe)
- Compile C++ code
- Link with libraries

	TPL2CPP		compiler		linker	
.TPL	==>	.CPP	==>	.OBJ	==>	.EXE

Go to example

Parameter file (MR.par)

```
# Number of parameters = 11 Objective function
value = 328.475 Maximum gradient component =
5.58429e-006
# estS:
0.718182 0.434671 0.478170 0.626118 0.598533
# p:
0.696203 0.923077 0.913043 0.900789 0.932414
0.530612
```

Standard deviation file (MR.std): results identical to Lebreton et al. 1992

```
index name value std dev
1 estS 7.1818e-001 1.5555e-001
2 estS 4.3467e-001 6.8829e-002
3 estS 4.7817e-001 5.9709e-002
4 estS 6.2612e-001 5.9266e-002
5 estS 5.9853e-001 5.6052e-002
6 p 6.9620e-001 1.6576e-001
7 p 9.2308e-001 7.2878e-002
8 p 9.1304e-001 5.8176e-002
9 p 9.0079e-001 5.3833e-002
10 p 9.3241e-001 4.5803e-002
11 p 5.3061e-001 5.0413e-002
```

Correlation Matrix (MR.COR)

The logarithm of the determinant of the hessian = 59.3873

index	name	value	std dev	1	2	3	...
1	estS	7.1818e-001	1.5555e-001	1.0000			
2	estS	4.3467e-001	6.8829e-002	-0.2067	1.0000		
3	estS	4.7817e-001	5.9709e-002	-0.0000	-0.0582	1.0000	
4	estS	6.2612e-001	5.9266e-002	0.0000	-0.0000	-0.0613	...
5	estS	5.9853e-001	5.6052e-002	-0.0000	-0.0000	-0.0000	...
6	p	6.9620e-001	1.6576e-001	-0.5638	0.1880	0.0000	...
7	p	9.2308e-001	7.2878e-002	-0.0000	-0.2619	0.1167	...
8	p	9.1304e-001	5.8176e-002	0.0000	-0.0000	-0.1980	...
9	p	9.0079e-001	5.3833e-002	0.0000	-0.0000	-0.0000	...
10	p	9.3241e-001	4.5803e-002	0.0000	-0.0000	-0.0000	...
11	p	5.3061e-001	5.0413e-002	0.0000	-0.0000	-0.0000	...

Report file (MR.rep)

```
S 0.718182 0.434671 0.47817 0.626118 0.598533
1
p 0.696203 0.923077 0.913043 0.900789 0.932414
0.530612
```

Random effects

$$p_t = \frac{\exp[\mu_p + \sigma_p \varepsilon_p]}{1 + \exp[\mu_p + \sigma_p \varepsilon_p]}$$

$$\varepsilon_p \sim N(0,1)$$

$$S_t = \frac{\exp[\mu_S + \sigma_S \varepsilon_S]}{1 + \exp[\mu_S + \sigma_S \varepsilon_S]}$$

$$\varepsilon_S \sim N(0,1)$$

PARAMETER_SECTION.

```
init_number meanS(1)
init_number ln_sdS(2)
init_number meanP(1)
init_number ln_sdP(2)
random_effects_vector Sdev(1,NRCperiods,2)
random_effects_vector pdev(1,NRCperiods,2)
vector S(1,NRCperiods)
vector p(1,NRCperiods)
number sdS
number sdP
matrix Scum(1,NRCperiods,1,NRCperiods)
matrix Pcum(1,NRCperiods,1,NRCperiods)
objective_function_value f
```

*.pin

```
# init_number meanS(1)
1
# init_number ln_sdS(2)
0
# init_number meanP(1)
1
# init_number ln_sdP(2)
0
# init_vector Sdev(1,NRCperiods,1)
0 0 0 0 0 0
# init_vector pdev(1,NRCperiods,1)
0 0 0 0 0 0
```

PROCEDURE_SECTION

```
PROCEDURE_SECTION
sdS=mfexp(ln_sds);
sdp=mfexp(ln_sdp);
S=elem_div(mfexp(meanS+Sdev*sdS),
(1+mfexp(meanS+Sdev*sdS)));
p=elem_div(mfexp(meancp+pdev*sdp),
(1+mfexp(meancp+pdev*sdp)));
for (int i=1;i<=NRCperiods;i++)
{
.
.
.
}
f+=0.5*norm2(Sdev)+0.5*norm2(pdev);
```

Run the model

Compilation

admb -re MR

Run

MR

For the penguins and albatross applications

MR -mno 2000

Standard deviations

```
sdreport_vector S(1,NRCperiods)
sdreport_vector p(1,NRCperiods)
```

Hierarchical Bayes

- Assume uniform priors on all parameters
 - Bounds on parameters are bounds on priors
 - Uniform priors are on the scale the parameter is estimated on (i.e. sd priors are uniform on log scale)

PROCEDURE_SECTION

```
if(mceval_phase())
{
ofstream
outsamples("samples.out",ios::app);
outsamples.precision(10);
outsamples<<meanS<<" "<<sdS<<
"<<meanp<<" "<<sdp<<" "<<s<<
"<<p<<endl;
outsamples.close();
}
```

Running model

```
MR -mcmc2 1000000 -mcsave 1000
MR -mceval
```

The two step approach provides the ability to change code
(e.g. change management strategy) without having to rerun
the MCMC procedure

Profile likelihood

```

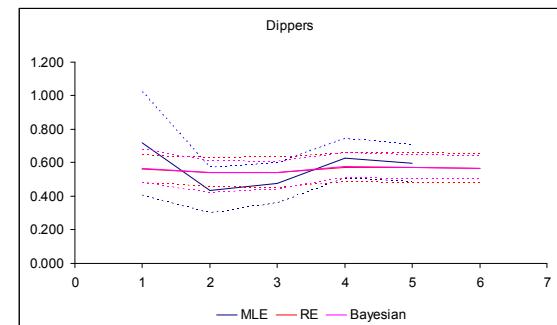
PARAMETER_SECTION
    likeprof_number Sprof1

PROCEDURE_SECTION
    Sprof1=S(1);

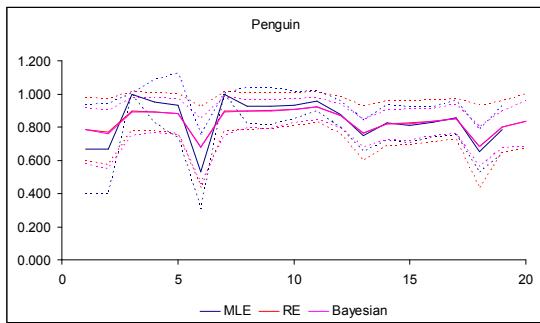
Running the model
    MR -lprof

```

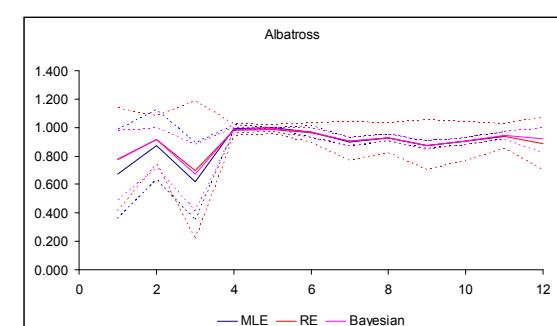
Dippers: Survival



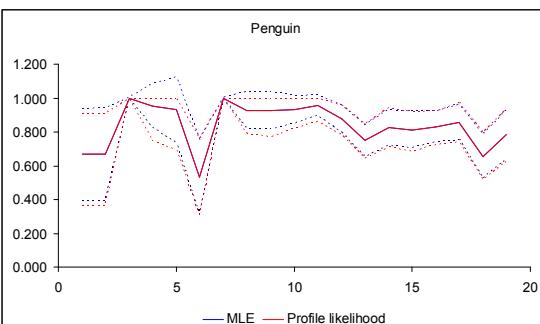
Penguins: Survival



Albatross: Survival



Profile likelihood: Survival



Random effects distribution parameters

	Dipper		Penguin		Albatross		
Smean	RE	Bayesian	RE	Bayesian	RE	Bayesian	
RE	0.241	0.244	1.741	1.772	2.604	3.181	
LB	0.016	-0.005	1.355	1.394	1.858	2.589	
UB	0.465	0.494	2.127	2.210	3.351	4.277	
Ssd	Estimate	0.120	0.133	1.286	0.718	1.123	1.582
LB	-0.246	0.012	0.502	0.304	0.418	0.757	
UB	0.486	0.520	2.070	1.288	1.829	4.259	
rmean	Estimate	2.223	2.284	1.729	1.742	1.224	1.161
LB	1.584	1.636	1.349	1.358	1.000	0.968	
UB	2.862	2.997	2.110	2.172	1.447	1.344	
rsd	Estimate	0.001	0.079	1.247	0.661	0.284	0.316
LB	-0.497	0.011	0.501	0.213	0.084	0.109	
UB	0.499	0.430	1.993	1.166	0.484	0.567	

Some disadvantages of ADMB

- Costs money
 - Memory constraints for Laplace approximation for large models

Further information

- ADMB website: <http://otter-rsch.com/admodel.htm> (includes demo version)
 - ADMB course: <http://www.iattc.org/iattc-staffMmaunderCourses-taught.htm>
 - ADMB list of papers:
<http://www.iattc.org/PDFFiles2/ADMBrferences.pdf>

Acknowledgements

- Richard Barker provided assistance on the algorithm for the mark-recapture model.
 - David Fournier provided information on AD Model builder, random effects, and statistical modeling in general
 - Data
 - Albatross: Jean-Dominic Lebreton and Sophie Veran
 - Penguin: John Darby and Dave Houston

