

# R documentation

## of 'KalmanEM.Rd'

January 27, 2009

---

KalmanEM

*Fitting Multivariate State Space Population Models (MSSMs)*

---

### Description

KalmanEM is a function to obtain maximum likelihood estimates of density dependence and spatial structure in multiple time series of single species population abundance, or to estimate interactions of multiple species.

### Usage

```
KalmanEM(y, whichPop = NA, Uinit = NA, Qinit = NA, Ainit = NA, Rinit = NA,
          x00 = NA, V00init = NA, max.iter = 5000, varcov.Q = "unconstrained",
          varcov.R = "diagonal", U.groups = NA, Q.groups = NA, R.groups = NA,
          U.bounds = c(-1, 1), logQ.bounds = c(log(1e-05), log(1)),
          logR.bounds = c(log(1e-05), log(1)), estInteractions = FALSE,
          MonteCarloInit = FALSE, numInits = 500, numInitSteps = 10,
          tol = 0.01, silent = FALSE)
```

### Arguments

y	Matrix (years x sites) of observed population abundances. If the algorithm is to be applied to log-abundance, the transformation should be done before being passed in. Missing values are allowed, but should be assigned the value -99.
whichPop	An optional vector used to assign sites to subpopulations (sites x 1)
Uinit	Initial value(s) for growth rate(s) U (subpopulations x 1)
Qinit	Initial value(s) for process error variance(s) (subpopulations x 1)
Ainit	Initial value(s) for offset (sites x 1 vector)
Rinit	Initial value(s) for observation error variance(s) (sites x 1)
x00	Initial value(s) for population state(s) (subpopulations x 1)
V00init	Variance of initial population states(s)
max.iter	Maximum number of iterations before the EM algorithm exits
varcov.Q	Structure of process error variance matrix: may be "unconstrained", "equalvarcov", or "diagonal"

<code>varcov.R</code>	Structure of observation error variance matrix: may be "unconstrained", "equal-varcov", or "diagonal"
<code>U.groups</code>	Vector (subpopulations x 1) used to allow shared growth rates between subpopulations
<code>Q.groups</code>	Vector (subpopulations x 1) used to allow shared process error variances between subpopulations, not applicable to the unconstrained cases.
<code>R.groups</code>	Vector (sites x 1) used to allow shared observation error variances between sites, not applicable to the unconstrained cases.
<code>U.bounds</code>	Optional constraints on growth rate(s)
<code>logQ.bounds</code>	Optional constraints on log of the process error variance(s)
<code>logR.bounds</code>	Optional constraints on log of the observation error variance(s)
<code>estInteractions</code>	Optional, allows matrix of density dependence terms (or interactions) to be estimated
<code>MonteCarloInit</code>	Should the parameter estimation be initialized from multiple random starting values?
<code>numInits</code>	Number of Monte Carlo random starting points
<code>numInitSteps</code>	Number of steps for each starting point
<code>tol</code>	Tolerance of improvement in the log-likelihood before the EM algorithm exist
<code>silent</code>	Should progress be printed to the user?

## Details

Data for KalmanEM may consist of time series of a single species from multiple sites ( $n$ ). The population dynamics of the sites may be independent, correlated, or the sites may represent repeated measurements of a single large panmictic population. Hypotheses about structure should be created from other available data (geography, DNA, diet, disease, etc). In addition to assigning sites to  $m$  subpopulations, the user has the option to allow for shared subpopulation (growth rate  $U$ , process error variance  $Q$ ) or site (observation error variance  $R$ ) parameters. Density dependence in each subpopulation may also be estimated ( $B$ ), but if not specified, subpopulations are assumed to be density independent. This function is also applicable to single time series, where all arguments to group structure are omitted.

The second application of KalmanEM is to time series of different species to estimate community interactions in a state-space framework. In this setting,  $n$  is the number of species, and for most scenarios, it should be assumed that they are independent ( $m = n$ ).

For both applications, it is strongly recommended that the `MonteCarloInit` option be set to TRUE (particularly if the model is not a good fit to the data). This requires more computation time, but reduces the chance of the algorithm becoming stuck and not reaching the true MLEs.

## Value

<code>states</code>	Matrix of MLEs of states, subpopulations x years
<code>A</code>	Vector of offsets for translating subpopulations to sites (the first site assigned to each subpopulation is assigned a value of 0, and not estimated.)
<code>B</code>	Matrix (subpopulations x subpopulations) of ML density dependence or interaction terms, if they are estimated

Q	Matrix (subpopulations x subpopulations) of ML process error variance estimates
R	Matrix (sites x sites) of ML observation error variance estimates
U	Vector (subpopulations x 1) of ML growth rate estimates
Z	Design matrix (sites x subpopulations) of 0s and 1s used to assign sites to subpopulations
x00	Vector (subpopulations x 1) of ML initial starting values
V00	Matrix (subpopulations x subpopulations) of variances, not estimated
V0T	Description of 'comp2'
Kt	Array (Subpopulations x sites x years) of Kalman gain, 0s for years without data
Innov	Matrix (Sites x years) of Kalman innovations, 0s for years without data
Sigma	Description of 'comp2'
m	Number of subpopulations, passed in to KalmanEM by user
n	Number of sites, passed in to KalmanEM by user
numYrs	Number of years, passed in to KalmanEM by user
iter	Number of EM iterations before the tolerance threshold was reached
loglike	Log-likelihood (maximum) when algorithm terminated
iter.record	A list, consisting of the following components stored at each iteration of the algorithm: U, array (subpopulations x iterations); Q, array (subpopulations x subpopulations x iterations); A, array (sites x iterations); R, array (sites x subpopulations x sites); loglike, vector (iterations x 1).
whichPop	Passed in by user, used by bootstrapMSSM
R.groups	Passed in by user, used by bootstrapMSSM
Q.groups	Passed in by user, used by bootstrapMSSM
U.groups	Passed in by user, used by bootstrapMSSM
varcov.R	Passed in by user, used by bootstrapMSSM
varcov.Q	Passed in by user, used by bootstrapMSSM
data	Passed in by user, used by bootstrapMSSM
include	Array (sites x subpopulations x years) of 0s and 1s, indicating whether data are missing (=0) or not (=1). This is later used by bootstrapMSSM.

### Author(s)

The R implementation of KalmanEM was done by Eli Holmes and Eric Ward at NOAA in Seattle, USA. Contact the authors at [eli\(dot\)holmes\(at\)noaa\(dot\)gov](mailto:eli(dot)holmes(at)noaa(dot)gov) or [eric\(dot\)ward\(at\)noaa\(dot\)gov](mailto:eric(dot)ward(at)noaa(dot)gov)

### References

1. ESA 1-day workshop files: <http://www.ecologybox.org>
2. Hinrichsen, R.A. and E.E. Holmes. 2008. Using multivariate state-space models to study spatial structure and dynamics. *To appear in: Emergent Challenges in Spatial Ecology* (eds S. Cantrell, C. Cosner, and S. Ruan).
3. Holmes, E.E., E.J. Ward, and R.A. Hinrichsen. 2009. Statistical methods for identifying spatial

structure within populations. *Submitted to Ecology*.

4. Shumway, R.H. and D.S. Stoffer. 2006. Time Series Analysis and Its Applications. Springer, New York.
5. Harvey, A.C. 1989. Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge University Press, Cambridge.
6. Cavanaugh, J.E. and R.H. Shumway. 1997. A bootstrap variant of AIC for state-space model selection. *Statistica* 7:473:496.

### See Also

Need to link to bootstrapMSSM here objects to See Also as [help](#),

### Examples

```
Need to come up with some simple examples + example datasets.
##----- Should be DIRECTLY executable !! -----
##-- ==> Define data, use random,
##-- or do help(data=index) for the standard data sets.
```

# R documentation

## of 'bootstrapMSSM.Rd'

January 27, 2009

---

bootstrapMSSM

*Bootstrapping Multivariate State Space Models (MSSMs)*

---

### Description

This function uses bootstrapping to generate confidence intervals on model parameters, and to calculate a bootstrapped version of AIC for model selection.

### Usage

```
bootstrapMSSM(nBoot, model, maxIter = 5000, tol = 0.01,  
              onlyAIC = TRUE, silent = FALSE)
```

### Arguments

nBoot	Number of bootstraps, should be more than 200.
model	A <code>KalmanEM</code> model object.
maxIter	Optional, the maximum number of iterations per bootstrap (default is the same as <code>KalmanEM</code> ).
tol	Optional, the tolerance per bootstrap (default is the same as <code>KalmanEM</code> ).
onlyAIC	Should only AICb be calculated? If not, confidence intervals are also generated.
silent	Should progress be printed to the user?

### Details

The function `bootstrapMSSM` should be called only after `KalmanEM` is used to return a `KalmanEM` model object. This model can then be passed to `bootstrapMSSM` to generate bootstrapped estimates of parameters and the bootstrapped version of AIC for model selection. The `bootstrapMSSM` function needs to be called for every candidate model, if AICb is going to be used as a model selection tool. If no missing data exists, the algorithm resamples the standardized innovations. In cases with missing data, the MLEs or  $R$  and  $Q$  are used to generate new datasets (with identical missing values to the original). The `KalmanEM` function is then initialized from the MLEs to create a parameter vector for the bootstrapped data, and upon termination, the function `Kfilter` is used to estimate the log-likelihood.

**Value**

AICb	Bootstrapped variant of AIC (smaller is better).
logL.star	Vector of log-likelihood values for each bootstrapped dataset.
U	If <code>onlyAIC = FALSE</code> , returns array of estimated growth rates (subpopulations x bootstraps).
Q	If <code>onlyAIC = FALSE</code> , returns array of estimated process error variances (subpopulations x subpopulations x bootstraps).
R	If <code>onlyAIC = FALSE</code> , returns array of estimated observation error variances (sites x sites x bootstraps).
A	If <code>onlyAIC = FALSE</code> , returns array of estimated offset parameters (sites x bootstraps).
B	If <code>onlyAIC = FALSE</code> , returns array of estimated interaction terms (subpopulations x subpopulations x bootstraps).
states	If <code>onlyAIC = FALSE</code> , returns array of estimated states (subpopulations x years x bootstraps).
boot.data	If <code>onlyAIC = FALSE</code> , returns array of bootstrapped datasets (years x sites x bootstraps).

**Author(s)**

The R implementation of KalmanEM was done by Eli Holmes and Eric Ward at NOAA in Seattle, USA. Contact the authors at `eli(dot)holmes(at)noaa(dot)gov` or `eric(dot)ward(at)noaa(dot)gov`

**References**

1. ESA 1-day workshop files: <http://www.ecologybox.org>
2. Hinrichsen, R.A. and E.E. Holmes. 2008. Using multivariate state-space models to study spatial structure and dynamics. *To appear in: Emergent Challenges in Spatial Ecology* (eds S. Cantrell, C. Cosner, and S. Ruan).
3. Holmes, E.E., E.J. Ward, and R.A. Hinrichsen. 2009. Statistical methods for identifying spatial structure within populations. *Submitted to Ecology*.
4. Shumway, R.H. and D.S. Stoffer. 2006. *Time Series Analysis and Its Applications*. Springer, New York.
5. Harvey, A.C. 1989. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, Cambridge.
6. Cavanaugh, J.E. and R.H. Shumway. 1997. A bootstrap variant of AIC for state-space model selection. *Statistica* 7:473:496.

**See Also**

Need to link to bootstrapMSSM here objects to See Also as [help](#),