

# R documentation

of all in ‘man’

March 31, 2022

## R topics documented:

fitness_boot . . . . .	1
manifold1Dplus . . . . .	4
scanner . . . . .	5
secondboot . . . . .	6
selection . . . . .	9

<b>Index</b>	<b>12</b>
--------------	-----------

---

fitness_boot	<i>fitness_boot</i>
--------------	---------------------

---

## Description

This function implements the first level of the parametric bootstrap procedure given by either Algorithm 1 or Algorithm 2 in Eck (2015) with respect to the mean-value parameterization. This is detailed in Steps 1 through 3d in the algorithm below. This parametric bootstrap generates resamples from the distribution evaluated at an envelope estimator of  $\tau$  adjusting for model selection volatility.

## Usage

```
fitness_boot(  
  model,  
  nboot,  
  index,  
  vectors = NULL,  
  dim = NULL,  
  data,  
  amat,  
  newdata,  
  modmat.new = NULL,  
  renewdata = NULL,  
  criterion = c("AIC", "BIC", "LRT"),  
  alpha = 0.05,  
)
```

```

fit.name = NULL,
method = c("eigen", "1d"),
quiet = FALSE,
corenum = NULL
)

```

### Arguments

model	An aster model object.
nboot	The number of bootstrap iterations desired.
index	The indices denoting which components of the canonical parameter vector are parameters of interest.
vectors	The indices denoting which reducing subspace of Fisher information is desired to construct envelope estimators. Must be specified if method = "eigen".
dim	The dimension of the envelope space used to construct envelope estimators. Must be specified if method = "1d".
data	An asterdata object corresponding to the original data.
amat	This object can either be an array or a matrix. It specifies a linear combination of mean-value parameters that correspond to expected Darwinian fitness. See the aster function help page in the original aster package for more details.
newdata	A dataframe corresponding to hypothetical individuals in which expected Darwinian fitness is to be estimated.
modmat.new	A model matrix corresponding to hypothetical individuals in which expected Darwinian fitness is to be estimated.
renewdata	A dataframe in long format corresponding to hypothetical individuals in which expected Darwinian fitness is to be estimated.
criterion	A model selection criterion of choice.
alpha	The type 1 error rate desired for the LRT.
fit.name	An expression that appears in the name of the nodes that correspond to Darwinian fitness. This is only necessary if renewdata is not provided.
method	The procedure used to obtain envelope estimators.
quiet	A logical argument. If FALSE, the function displays how much time it takes to run m iterations.
corenum	The number of cores specified for speeding up the bootstrap process.

### Details

This function implements the first level of the parametric bootstrap procedure given by either Algorithm 1 or Algorithm 2 in Eck (2015) with respect to the mean-value parameterization. This is detailed in Steps 1 through 3d in the algorithm below. This parametric bootstrap generates re-samples from the distribution evaluated at an envelope estimator of  $\tau$  adjusting for model selection volatility. The user specifies a model selection criterion which selects vectors that construct envelope estimators using the reducing subspace approach. The user also specifies which method is to be used in order to calculate envelope estimators. When one is using a partial envelope, then this function constructs envelope estimators of  $v$  where we write  $\tau = (\gamma^T, v^T)^T$  and  $v$  corresponds

to aster model parameters of interest. In applications, candidate reducing subspaces are indices of eigenvectors of  $\hat{\Sigma}_{v,v}$  where  $\hat{\Sigma}_{v,v}$  is the part of  $\hat{\Sigma}$  corresponding to our parameters of interest. These indices are specified by vectors. When all of the components of  $\tau$  are components of interest, then we write  $\hat{\Sigma}_{v,v} = \hat{\Sigma}$ . When data is generated via the parametric bootstrap, it is the indices (not the original reducing subspaces) that are used to construct envelope estimators constructed using the generated data. The algorithm using reducing subspaces is as follows:

1. Fit the aster model to the data and obtain  $\hat{\tau} = (\hat{\gamma}^T, \hat{v}^T)$  and  $\hat{\Sigma}$  from the aster model fit.
2. Compute the envelope estimator of  $v$  in the original sample, given as  $\hat{v}_{env} = P_{\hat{G}}\hat{v}$  where  $P_{\hat{G}}$  is computed using reducing subspaces and selected via a model selection criterion of choice.
3. Perform a parametric bootstrap by generating resamples from the distribution of the aster submodel evaluated at  $\hat{\tau}_{env} = (\hat{\gamma}^T, \hat{v}_{env}^T)^T$ . For iteration  $b = 1, \dots, B$  of the procedure:
  - (a) Compute  $\hat{\tau}^{(b)}$  and  $\hat{\Sigma}_{v,v}^{(b)}$  from the aster model fit to the resampled data.
  - (b) Build  $P_{\hat{G}}^{(b)}$  using the indices of  $\hat{\Sigma}_{v,v}^{(b)}$  that are selected using the same model selection criterion as Step 2 to build  $\hat{G}$ .
  - (c) Compute  $\hat{v}_{env}^{(b)} = P_{\hat{G}}^{(b)}\hat{v}^{(b)}$  and  $\hat{\tau}_{env}^{(b)} = (\hat{\gamma}^{(b)T}, \hat{v}_{env}^{(b)T})^T$ .
  - (d) Store  $\hat{\tau}_{env}^{(b)}$  and  $g(\hat{\tau}_{env}^{(b)})$  where  $g$  maps  $\tau$  to the parameterization of Darwinian fitness.
4. After  $B$  steps, the bootstrap estimator of expected Darwinian fitness is the average of the envelope estimators stored in Step 3d. This completes the first part of the bootstrap procedure.
5. We now proceed with the second level of bootstrapping at the  $b^{th}$  stored envelope estimator  $\hat{\tau}_{env}^{(b)}$ . For iteration  $k = 1, \dots, K$  of the procedure:
  - (a) Generate data from the distribution of the aster submodel evaluated at  $\hat{\tau}_{env}^{(b)}$ .
  - (b) Perform Steps 3a through 3d with respect to the dataset obtained in Step 5a.
  - (c) Store  $\hat{\tau}_{env}^{(b)(k)}$  and  $g(\hat{\tau}_{env}^{(b)(k)})$ .

The parametric bootstrap procedure which uses the 1d algorithm to construct envelope estimators is analogous to the above algorithm. To use the 1d algorithm, the user specifies method = "1d". A parametric bootstrap generating resamples from the distribution evaluated at the aster model MLE is also conducted by this function.

## Value

a list containing the following elements in order:

env.boot.out	Estimated expected Darwinian fitness from generated data obtained from Steps 3a-3d in the bootstrap procedure using the envelope estimator constructed using reducing subspaces.
MLE.boot.out	Estimated expected Darwinian fitness from generated data obtained from Steps 3a-3d in the bootstrap procedure using the MLE.
MLE.tau.boot	Estimated mean-value parameter vectors from generated data obtained from Steps 3a-3d in the bootstrap procedure using the MLE.
env.tau.boot	Estimated mean-value parameter vectors from generated data obtained from Steps 3a-3d in the bootstrap procedure using the envelope estimator constructed using the 1d algorithm.
P.list	A list of all estimated projections into the envelope space constructed from reducing subspaces for Steps 3a-3d in the bootstrap procedure.
vectors.list	A list of indices of eigenvectors used to build the projections in P.list. These indices are selected using the user specified model selection criterion as indicated in Steps 3a-3d in the bootstrap procedure.

## References

- Cook, R.D. and Zhang, X. (2014). Foundations for Envelope Models and Methods. *JASA*, In Press.
- Cook, R.D. and Zhang, X. (2015). Algorithms for Envelope Estimation. *Journal of Computational and Graphical Statistics*, Published online. doi: [10.1080/10618600.2015.1029577](https://doi.org/10.1080/10618600.2015.1029577).
- Eck, D. J., Geyer, C. J., and Cook, R. D. (2016). Enveloping the aster model. *in prep*.
- Eck, D.~J., Geyer, C.~J., and Cook, R.~D. (2016). Web-based Supplementary Materials for “Enveloping the aster model.” *in prep*.
- Efron, B. (2014). Estimation and Accuracy After Model Selection. *JASA*, **109:507**, 991-1007.

## Examples

```
## Not run: # see Web-based Supplementary Materials for ``Enveloping the aster model.``
```

---

manifold1Dplus	<i>manifold1Dplus</i>
----------------	-----------------------

---

## Description

The 1D algorithm

## Usage

```
manifold1Dplus(M, U, u)
```

## Arguments

- |   |   |
|---|---|
| M | A $\sqrt{n}$ estimate of an estimator's asymptotic covariance matrix.   |
| U | A $\sqrt{n}$ estimate of the parameter associated with the space we are enveloping. For our purposes this quantity is either the outer product of the MLE of the mean-value submodel parameter vector with itself or the outer product of the MLE of the canonical submodel parameter vector with itself. |
| u | The dimension of the envelope space assumed.  |

## Details

This function calls `get1Dobj`, `get1Dini`, and `get1Dderiv` in order to find

$$\max_w [\log(w^T M w) + \log(w^T (M + U) w) - 2 \log(w^T w)]$$

using Polak-Ribiere conjugate gradient in `optim`. This maximization is conducted a total of `u` times and at each iteration a vector belonging to the envelope space is returned. The vector returned at a specific iteration is orthogonal to the vectors returned at previous iterations. When finished, a basis matrix for the envelope space is returned.

## Value

- |   |  |
|---|--|
| G | A $\sqrt{n}$ estimator of the basis matrix for the envelope subspace. This matrix has <code>u</code> columns |
|---|--|

## References

Cook, R.D. and Zhang, X. (2014). Foundations for Envelope Models and Methods. *JASA*, In Press.

Cook, R.D. and Zhang, X. (2015). Algorithms for Envelope Estimation. *Journal of Computational and Graphical Statistics*, Published online. doi: [10.1080/10618600.2015.1029577](https://doi.org/10.1080/10618600.2015.1029577).

## Examples

```
## Not run: library(envlpaster)
data(simdata30nodes)
data <- simdata30nodes.asterdata
nnode <- length(vars)
xnew <- as.matrix(simdata30nodes[,c(1:nnode)])
m1 <- aster(xnew, root, pred, fam, modmat)
avar <- m1$fisher
beta <- m1$coef
U <- beta \
manifold1Dplus(M = avar, U = U, u = 1)
## End(Not run)
```

---

 scanner

 scanner
 

---

## Description

A diagnostic assessing the potential benefits of envelope methodology for a particular aster model.

## Usage

```
scanner(M, coef, u)
```

## Arguments

M	A $\sqrt{n}$ estimate of an estimator's asymptotic covariance matrix.
coef	The MLE of the parameter of interest.
u	The dimension of the envelope space assumed.

## Details

This function provides users with a rough diagnostic for the performance of an envelope estimator at a specific dimension. We can see how close a particular potential envelope estimator is to the MLE as well as the proportion of variation that will be discarded when using envelope estimation. This amount of variation discarded is optimistic since it does not account for variability associated with estimating the projection into the envelope space.

## Value

indices	The indices of the u most relevant eigenvectors of M to the construction of coef.
table	An output table. The first column is the projection of coef into the space spanned by the eigenvectors given by the indices.
G	The u most relevant eigenvectors of M to the construction of coef.
prop	the sum of the remaining eigenvalues of M divided by the sum of all of the eigenvalues of M.

## Examples

```
## Not run: library(envlpaster)
data(simdata30nodes)
data <- simdata30nodes.asterdata
nnode <- length(vars)
xnew <- as.matrix(simdata30nodes[,c(1:nnode)])
m1 <- aster(xnew, root, pred, fam, modmat)
avar <- m1$fisher
beta <- m1$coef
scanner(M = avar, coef = beta, u = 1)
## End(Not run)
```

---

secondboot

*secondboot*


---

## Description

A parametric bootstrap procedure evaluated at an envelope estimator of the submodel mean-value parameter vector  $\tau$  that was obtained using eigenstructures or the 1d algorithm.

## Usage

```
secondboot(
  k,
  nboot2,
  out,
  model,
  index,
  data,
  amat,
  newdata,
  method = c("eigen", "1d")
)
```

## Arguments

k	The index of the top level parametric bootstrap procedure conducted by fit.boot.Efron that the second level of bootstrapping is being applied to.
nboot2	The bootstrap sample size for the second level of parametric bootstrapping.
out	The output of fit.boot.Efron.
model	An aster model object.
index	The indices denoting which components of the canonical parameter vector are parameters of interest.
data	An asterdata object corresponding to the original data.
amat	This object can either be an array or a matrix. It specifies a linear combination of mean-value parameters that correspond to expected Darwinian fitness. See the aster function help page in the original aster package for more details.
newdata	A dataframe corresponding to hypothetical individuals in which expected Darwinian fitness is to be estimated.
method	The procedure used to obtain envelope estimators.

## Details

This function implements the second level of the parametric bootstrap procedure given by either Algorithm 1 or Algorithm 2 in Eck (2015) with respect to the mean-value parameterization. This is detailed in Steps 4 through 5c in the algorithm below. At iteration  $b$ , this parametric bootstrap generates resamples from the distribution evaluated at the envelope estimator ( $\hat{\tau}_{env}^{(b)}$ ) of  $\tau$ . In this case, the selected indices producing the eigenstructure which was used to construct the envelope estimator  $\hat{\tau}_{env}^{(b)}$  are used to construct envelope estimators for the generated data. These resamples are used to estimate the variability of  $\hat{\tau}_{env}^{(b)}$ . The algorithm using eigenstructures is as follows:

1. Fit the aster model to the data and obtain  $\hat{\tau} = (\hat{\gamma}^T, \hat{v}^T)$  and  $\hat{\Sigma}$  from the aster model fit.
2. Compute the envelope estimator of  $v$  in the original sample, given as  $\hat{v}_{env} = P_{\hat{G}}\hat{v}$  where  $P_{\hat{G}}$  is computed using eigenstructures and selected via a model selection criterion of choice.
3. Perform a parametric bootstrap by generating resamples from the distribution of the aster submodel evaluated at  $\hat{\tau}_{env} = (\hat{\gamma}^T, \hat{v}_{env}^T)^T$ . For iteration  $b = 1, \dots, B$  of the procedure:
  - (a) Compute  $\hat{\tau}^{(b)}$  and  $\hat{\Sigma}_{v,v}^{(b)}$  from the aster model fit to the resampled data.
  - (b) Build  $P_{\hat{G}}^{(b)}$  using the indices of  $\hat{\Sigma}_{v,v}^{(b)}$  that are selected using the same model selection criterion as Step 2 to build  $\hat{G}$ .
  - (c) Compute  $\hat{v}_{env}^{(b)} = P_{\hat{G}}^{(b)}\hat{v}^{(b)}$  and
  - (d) Store  $\hat{\tau}_{env}^{(b)}$  and  $g(\hat{\tau}_{env}^{(b)})$  where  $g$  maps  $\tau$  to the parameterization of Darwinian fitness.
4. After  $B$  steps, the bootstrap estimator of expected Darwinian fitness is the average of the envelope estimators stored in Step 3d. This completes the first part of the bootstrap procedure.
5. We now proceed with the second level of bootstrapping at the  $b^{th}$  stored envelope estimator  $\hat{\tau}_{env}^{(b)}$ . For iteration  $k = 1, \dots, K$  of the procedure:
  - (a) Generate data from the distribution of the aster submodel evaluated at  $\hat{\tau}_{env}^{(b)}$ .
  - (b) Perform Steps 3a through 3d with respect to the dataset obtained in Step 5a.
  - (c) Store  $\hat{\tau}_{env}^{(b)(k)}$  and  $g(\hat{\tau}_{env}^{(b)(k)})$ .

When the second level of bootstrapping is completed for all  $b = 1, \dots, B$  then this function reports the standard deviation of the bootstrapped envelope estimator of expected Darwinian fitness. In this case, the bootstrap procedure accounts for model selection volatility. The bootstrapped envelope estimator is

$$\hat{\mu}_g = \frac{1}{B} \sum_{b=1}^B g(\hat{\tau}_{env}^{(b)})$$

where  $g(\hat{\tau}_{env}^{(b)})$  are the stored envelope estimators of expected Darwinian fitness in the `env.boot.out` matrix included in the output of `fit.boot.Efron`. The standard deviation of the bootstrapped envelope estimator of expected Darwinian fitness is

$$\sum_{b=1}^B \left[ \widehat{cov}^{(b)T} \hat{V}^{-1} \widehat{cov}^{(b)} \right] / B$$

where  $\widehat{cov}^{(b)} = \mathbf{B}^{(b)T} C^{(b)} / K$  and  $\hat{V} = \mathbf{B}^{(b)T} \mathbf{B}^{(b)} / K$ . The matrix  $\mathbf{B}^{(b)} \in R^{K \times p}$  has rows given by

$$\hat{\tau}_{env}^{(b)(k)} - \sum_{k=1}^K \hat{\tau}_{env}^{(b)(k)} / K$$

and the matrix  $C^{(b)} \in R^{K \times d}$  has columns given by

$$g\left(\tau_{env}^{(b)(k)}\right) - g\left(\tau_{env}^{(b)}\right)$$

.

For more details, see Efron (2014) and Eck (2015). The parametric bootstrap procedure which uses the 1d algorithm to construct envelope estimators is analogous to the above algorithm. To use the 1d algorithm, the user specifies `method = "1d"` instead of `method = "eigen"`.

## Value

<code>sd.Efron</code>	The estimated standard deviation (sd) for estimated expected Darwinian fitness where is estimation is conducted using envelope methodology. This sd accounts for model selection volatility. An eigenvalue decomposition using eigen is used internally to calculate this quantity.
<code>cov</code>	A components needed to construct <code>sd.Efron</code> if other numerical methods are desired.
<code>V</code>	A components needed to construct <code>sd.Efron</code> if other numerical methods are desired.
<code>MLE.tau.boot.subsample</code>	A components needed to construct <code>sd.Efron</code> if other numerical methods are desired.
<code>est.env.subsample</code>	A components needed to construct <code>sd.Efron</code> if other numerical methods are desired.

## References

- Cook, R.D. and Zhang, X. (2014). Foundations for Envelope Models and Methods. *JASA*, In Press.
- Cook, R.D. and Zhang, X. (2015). Algorithms for Envelope Estimation. *Journal of Computational and Graphical Statistics*, Published online. doi: [10.1080/10618600.2015.1029577](https://doi.org/10.1080/10618600.2015.1029577).
- Eck, D. J., Geyer, C. J., and Cook, R. D. (2016). Enveloping the aster model. *in prep*.
- Eck, D.~J., Geyer, C.~J., and Cook, R.~D. (2016). Web-based Supplementary Materials for “Enveloping the aster model.” *in prep*.
- Efron, B. (2014). Estimation and Accuracy After Model Selection. *JASA*, **109:507**, 991-1007.

## Examples

```
### Web-based Supplementary Materials for ``Enveloping the aster model.'' ###
```



---

selection	<i>selection</i>
-----------	------------------

---

## Description

Finds the envelope eigenspace or dimension that is favored using AIC, BIC, or the LRT at a specified size.

## Usage

```
selection(
  parm,
  index,
  model,
  data = NULL,
  alpha = 0.05,
  type = c("canonical", "mean-value"),
  method = c("eigen", "1d")
)
```

## Arguments

parm	The MLE of the parameter of interest.
index	The indices denoting which components of the canonical parameter vector are parameters of interest.
model	An aster model object.
data	An asterdata object corresponding to the original data.
alpha	The type 1 error rate desired for the LRT.
type	The parameterization of the aster model in which envelope methods are being applied.
method	The procedure used to obtain envelope estimators.

## Details

This function provides the user with the envelope model dimension or indices of the eigenspace favored by AIC, BIC, and the likelihood ratio test of size  $\alpha$ . There are four possible combinations of outputs. They are:

1. The specification of `method = "eigen"` and `type = "mean-value"` provides the user with the indices of the eigenspace of estimated Fisher information used to construct an envelope estimator for  $\tau$  favored by AIC, BIC, and the LRT of size  $\alpha$ .
2. The specification of `method = "eigen"` and `type = "canonical"` provides the user with the indices of the eigenspace of estimated Fisher information used to construct an envelope estimator for  $\beta$  favored by AIC, BIC, and the LRT of size  $\alpha$ .
3. The specification of `method = "1d"` and `type = "mean-value"` provides the user with the envelope model dimension used to construct an envelope estimator for  $\tau$  favored by AIC, BIC, and the LRT of size  $\alpha$ .
4. The specification of `method = "1d"` and `type = "canonical"` provides the user with the envelope model dimension used to construct an envelope estimator for  $\beta$  favored by AIC, BIC, and the LRT of size  $\alpha$ .

When one is interested in envelope model dimensions or eigenspaces with respect to  $\beta$ , then an `asterdata` object does not need to be specified. On the other hand, an `asterdata` is needed in order to map the estimated  $\tau$  to its corresponding  $\beta$  value. This is necessary because of the interface (or lack thereof) between current `aster` and `aster2` software. The way in which `aster` model log likelihoods are evaluated is incorporated in `aster` software and changing parameterizations is carried out using `aster2` software.

### Value

<code>parm</code>	The MLE of the parameter of interest.
<code>index</code>	The indices denoting which components of the canonical parameter vector are parameters of interest.
<code>model</code>	An <code>aster</code> model object.
<code>data</code>	An <code>asterdata</code> object.
<code>alpha</code>	The desired size of the LRT.
<code>type</code>	The parameterization of the <code>aster</code> model in which envelope methods are being applied.
<code>method</code>	The procedure used to obtain envelope estimators.

### References

- Cook, R.D. and Zhang, X. (2014). Foundations for Envelope Models and Methods. *JASA*, In Press.
- Cook, R.D. and Zhang, X. (2015). Algorithms for Envelope Estimation. *Journal of Computational and Graphical Statistics*, Published online. doi: [10.1080/10618600.2015.1029577](https://doi.org/10.1080/10618600.2015.1029577).
- Eck, D. J., Geyer, C. J., and Cook, R. D. (2016). Enveloping the `aster` model. *in prep*.

### Examples

```
## Not run:
set.seed(13)
library(envlpaster)
library(aster2)
data(generateddata)
m.null <- aster(resp ~ 0 + varb, fam = fam, pred = pred,
               varvar = varb, idvar = id, root = root, data = redata)
m1 <- aster(resp ~ 0 + varb + mass + timing,
            fam = fam, pred = pred, varvar = varb, idvar = id,
            root = root, data = redata)
m2 <- aster(resp ~ 0 + varb + mass + timing +
            I(mass^2) + I(timing^2) + I(mass*timing),
            fam = fam, pred = pred, varvar = varb, idvar = id,
            root = root, data = redata)
anova.table <- anova(m.null,m1,m2); anova.table
beta <- m1$coef
a <- grepl( "offsp", names(beta))
a <- a + grepl( "surviv", names(beta))
b <- which(a == 1)
target <- c(1:length(beta))[-b]
nnode <- ncol(m1$x)
data.aster <- asterdata(data, vars, pred, rep(0,nnode),
                       fam, families = list("bernoulli", "poisson",
```

```
                                fam.zero.truncated.poisson()))
selection(parm = beta, index = target, model = m1,
          data = data.aster, alpha = 0.05, type = "canonical",
          method = "eigen")

## End(Not run)
```

# Index

fitness\_boot, [1](#)  
manifold1Dplus, [4](#)  
scanner, [5](#)  
secondboot, [6](#)  
selection, [9](#)