

Package ‘selectiongain’

February 15, 2012

Type Package

Title Expected gain from multistages selection

Version 1.2.1.0

Date 2011-12-07

Author Xuefei Mi, Friedrich Utz, Albrecht E. Melchinger (Project contact person: Melchinger at uni-hohenheim.de)

Maintainer Xuefei Mi <mi@pz.uni-hohenheim.de>

Depends R (>= 2.13.0), mvtnorm

Description This package calculate the gain from selection, which is described by Cochran (1951). For one-stage selection the gain is defined as $\Delta G(y) = i \rho_y \rho_1$, where i is the selection intensity, ρ_1 is the correlation between the true breeding value and the selection index y (Utz1969). The numerical calculation is based on Tallis’ algorithm (Tallis1961).

License GPL-2

Repository CRAN

Date/Publication 2011-12-10 11:13:45

R topics documented:

calculatefromalpha	2
multistagegain	4
multistagegain.each	7
multistageselection	12
multistagevariance	15

Index 21

calculatefromalpha *Non-equi coordinate quantile for singly-truncated multivariate normal distribution*

Description

This function calculates the non-equi coordinate quantile vector $Q = \{q_1, q_2, \dots, q_n\}$ for a multivariate normal distribution from a given alpha vector $\vec{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$. It can be compared with the function `qmvnorm()` in R-package **mvtnorm**, which calculates the equi coordinate quantile q for multi-variate normal distribution from a given alpha scalar.

E.g. Let f be the density function, for $n = 2$:

the equi coordinate quantile $\vec{q} = \{q, q, \dots, q\}$ defined by **mvtnorm** is:

$$\alpha = \int_q^\infty \int_q^\infty f(x_1, x_2) dx_1 dx_2$$

the non-equi coordinate quantile $Q = \{q_1, q_2, \dots, q_n\}$ defined by us is:

$$\alpha_1 = \int_{q_1}^\infty f(x_1) dx_1$$

$$\alpha_2 = \int_{q_1}^\infty \int_{q_2}^\infty f(x_1, x_2) dx_1 dx_2$$

Usage

```
calculatefromalpha(alpha, dim, corr, alg)
```

Arguments

<code>dim</code>	Dimension of the random variable X.
<code>alpha</code>	Probability vector $\vec{\alpha}$ for random variable X. In plant breeding, it is also called the selected fraction.
<code>corr</code>	Correlation matrix
<code>alg</code>	It decides which algorithm will be used, the Genz and Bretz' algorithm is used by default. The Miwa's algorithm can be optional (Mi. et al. 2009; Genz. et al. 2010)

Details

There is a limitation, if the absolute value of q_i is greater than 100, then the absolute value is settled to be 100.

Value

The out come is a vector for the coordinates.

Note

When a vector $\vec{\alpha}$ is given, the quantiles are calculated consecutively to satisfy the given $\vec{\alpha}$. The calculation from other direction to minus infinite of the integral is also possible for `qmvnorm()`. But not for this function, the user have to reverse the integral.

Author(s)

Xuefei Mi

References

W.G. Cochran. Improvent by means of selection. In: Proceedings Second Berkeley Symposium on Math Stat Prof, pp449-470.

G.M. Tallis. Moment generating function of truncated multi-normal distribution. Journal of the Royal Statistical Society, Series B, 23(1):223-229, 1961.

H.F. Utz. Mehrstufenselektion in der Pflanzenzuechtung. Doctor thesis, University Hohenheim, 1969.

X. Mi, T. Miwa and T. Hothorn. Implement of Miwa's analytical algorithm of multi-normal distribution, R Journal, 1:37-39, 2009.

A., Genz, F., Bretz (2009). Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics, Vol. 195, Springer-Verlag, Heidelberg.

A., Genz, F., Bretz, T., Miwa, X., Mi, F., Leisch, F., Scheipl, T., Hothorn, 2010. mvtnorm: Multivariate normal and t distributions. R package version 0.9-9.

See Also

selectiongain(), qnorm()

Examples

```
##---- Should be DIRECTLY executable !! ----  
##-- ==> Define data, use random,  
##--or do help(data=index) for the standard data sets.
```

```
dim=3  
alpha=rep(0,dim)  
  
alpha[1] = 0.3333  
alpha[2] = 0.25  
alpha[3] = 0.2
```

```
corr=diag(3)
```

```
corr12=0.3016  
corr[1,2]=corr12  
corr[2,1]=corr12
```

```
corr13=0.5630  
corr[1,3]=corr13  
corr[3,1]=corr13
```

```
corr23=0.5630
```

```

corr[2,3]=corr23
corr[3,2]=corr23

calculatefromalpha(alpha,dim=3,corr)

```

multistagegain *Expected gain for k-stages selection*

Description

This function calculates the well-known selection gain $\Delta G(y)$, which is described by Cochran (1951). For one-stage selection the gain is defined as $\Delta G(y) = i\rho_y\rho_1$, where i is the selection intensity, ρ_1 is the correlation between the true breeding value and the selection index y (Utz 1969).

During a selection procedure, questions such as "How great is the yield after selection?" and "After three years testing and selection, how much is the chance that we miss one good candidate or the best candidate?" are frequently asked. These two questions can be answered by calculating the expected gain and variance of the gain, of a truncated multi-normal distribution (Tallis 1961). A numerical equation for calculating such moment generating function is since a long time available (Utz 1969). However, the properties of this old algorithm is limited. It is slow and can only handle a restricted selection model, which has at most three independent variables.

We developed an R-package for maximizing the gain of a multi-stage selection procedure under certain restrictions, e.g., a given annual budget or certain risk limits of each stage. This package is applied in the fields of plant/animal breeding, where a multi-normal regression model is commonly built. It can handle a restricted selection model with up to eight independent variables within seconds.

Usage

```
multistagegain(k, corr, alphaofx, sum.dim,alg,stages=FALSE)
```

Arguments

sum.dim	It is the dimension of k, which is dimension of x plus 1 (y has one dimension).
k	The lower bound of the integral, should be a vector. The first element of k is the lower bound of the response variable, Y, which is usually set as -Inf. The dimension of k should be the dimension of selection stage plus one.
corr	correlation matrix.
alphaofx	Selected fraction, this is a scalar not a vector.
alg	It decides which algorithm will be used, the Genz and Brett' algorithm is used by default. The Miwa's algorithm can be optional (Mi. et al. 2009; Genz. et al. 2010).
stages	If = TRUE, the gain from each stages will be shown

Details

There are only three functions in the package. The functions are used to calculate the moment generating function of the truncated multivariate normal distribution. The selection gain can be described as follow:

$x_i = y + e_i$, where x_i : observed mean value of the character at the i th stage, y : the true genetic value which generates the observation, e_i : experiment error, which is assumed to be normally distributed. The true genetic value can be estimated by the observations with a regression function, $y(x) = f(x_1, x_2, \dots, x_i)$. A fraction u is defined as the ratio of the variance of e_1 and y

Under the normality assumption, y and x_i are multivariate normally distributed. We want to maximize the expected value of y in the restricted area, which is specified by the selection fraction. The Moment Generating Function (MGF) is used to calculate this multi-dimensional integral (Tallis, 1961; Utz, 1969).

Computation of the expected value of y requires computation of the following multi-dimensional integral over the restricted area defined by the selection. Let $\Omega = \{x_i \geq u_i; 1 \leq i \leq k\}$, $U = \{u_1, \dots, u_k\}$, ϕ_k : the density function of the multivariate normal distribution. The integral is given by $E_{\Omega}(y) = \int_{-\infty}^{\infty} y(x) \int_0^{\infty} \dots \int_0^{\infty} \phi_k(X; U, \Sigma) dx_1 \dots dx_k dy$

For further details about the whole project (selection gain for Double haploids and selection gain with markers), please contact the project contact person. The most important function is "multistageselection", which calculates the selection gain from given selection fraction and correlation matrix.

Value

The value returned, is the gain of selection.

Note

No further notes

Author(s)

Xuefei Mi, H. Friedrich Utz, Albrecht E. Melchinger

Maintainer: Xuefei Mi, <mi at pz.uni-hohenheim.de>

Contact person: Albrecht E. Melchinger, melchinger at uni-hohenheim.de

References

W.G. Cochran. Improvement by means of selection. In: Proceedings Second Berkeley Symposium on Math Stat Prof, pp449-470, 1951

G.M. Tallis. Moment generating function of truncated multi-normal distribution. Journal of the Royal Statistical Society, Series B, 23(1):223-229, 1961.

H.F. Utz. Mehrstufenselektion in der Pflanzenzuechtung. Doctor thesis, University Hohenheim, 1969.

X. Mi, T. Miwa and T. Hothorn. Implement of Miwa's analytical algorithm of multi-normal distribution, R Journal, 1:37-39, 2009.

A., Genz, F., Bretz. Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics, Vol. 195, Springer-Verlag, Heidelberg, 2009

A., Genz, F., Bretz, T., Miwa, X., Mi, F., Leisch, F., Scheipl, T., Hothorn. mvtnorm: Multivariate normal and t distributions. R package version 0.9-9, 2010.

See Also

No link

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

corr=diag(4)

corr12=0.3508
corr[1,2]=corr12
corr[2,1]=corr12

corr13=0.3508
corr[1,3]=corr13
corr[3,1]=corr13

corr14=0.4979
corr[1,4]=corr14
corr[4,1]=corr14

corr23=0.3016
corr[2,3]=corr23
corr[3,2]=corr23

corr24=0.5630
corr[2,4]=corr24
corr[4,2]=corr24

corr34=0.5630
corr[3,4]=corr34
corr[4,3]=corr34

quantile=c(0.4308,0.9804,1.8603)
k=c(-200,quantile)
# -200 is small enough to send the lower bound to be -Inf

alphaofx=pmvnorm(lower=c(-200,quantile),corr=corr)

multistagegain(k=c(-200,quantile),corr=corr,alphaofx,sum.dim=4)
```

```
#####
# new code for adapt the window input of the red-R
#####

k=c(-200,0.4308,0.9804,1.8603)

corr=matrix( c(1,      0.3508,0.3508,0.4979,
               0.3508 ,1,      0.3016,0.5630,
               0.3508, 0.3016,1      ,0.5630,
               0.4979, 0.5630,0.5630,1),
             nrow=4
           )

sum.dim=4

alphaofx=pmvnorm(lower=k,corr=corr)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=4)

#####
# new code for testing the parameter stages
#####

multistagegain(k=k,corr=corr,alphaofx,sum.dim=4,stages=TRUE)
```

multistagegain.each *Expected gain for k-stages selection*

Description

This function calculates the well-known selection gain $\Delta G(y)$, which is described by Cochran (1951). For one-stage selection the gain is defined as $\Delta G(y) = i\rho_y\rho_1$, where i is the selection intensity, ρ_1 is the correlation between the true breeding value and the selection index y (Utz 1969).

During a selection procedure, questions such as "How great is the yield after selection?" and "After three years testing and selection, how much is the chance that we miss one good candidate or the best candidate?" are frequently asked. These two questions can be answered by calculating the expected gain and variance of the gain, of a truncated multi-normal distribution (Tallis 1961). A numerical equation for calculating such moment generating function is since a long time available (Utz 1969). However, the properties of this old algorithm is limited. It is slow and can only handle a restricted selection model, which has at most three independent variables.

We developed an R-package for maximizing the gain of a multi-stage selection procedure under certain restrictions, e.g., a given annual budget or certain risk limits of each stage. This package

is applied in the fields of plant/animal breeding, where a multi-normal regression model is commonly built. It can handle a restricted selection model with up to eight independent variables within seconds.

Usage

```
multistagegain.each(k, corr, alphaofx, sum.dim,alg)
```

Arguments

sum.dim	It is the dimension of k, which is dimension of x plus 1 (y has one dimension).
k	The lower bound of the integral, should be a vector. The first element of k is the lower bound of the response variable, Y, which is usually set as -Inf. The dimension of k should be the dimension of selection stage plus one.
corr	correlation matrix.
alphaofx	Selected fraction, this is a scalar not a vector.
alg	It decides which algorithm will be used, the Genz and Brett' algorithm is used by default. The Miwa's algorithm can be optional (Mi. et al. 2009; Genz. et al. 2010).

Details

There are only three functions in the package. The functions are used to calculate the moment generating function of the truncated multivariate normal distribution. The selection gain can be described as follow:

$x_i = y + e_i$, where x_i : observed mean value of the character at the i th stage, y : the true genetic value which generates the observation, e_i : experiment error, which is assumed to be normally distributed. The true genetic value can be estimated by the observations with a regression function, $y(x) = f(x_1, x_2, \dots, x_i)$. A fraction α is defined as the ratio of the variance of e_1 and y

Under the normality assumption, y and x_i are multivariate normally distributed. We want to maximize the expected value of y in the restricted area, which is specified by the selection fraction. The Moment Generating Function (MGF) is used to calculate this multi-dimensional integral (Tallis, 1961; Utz, 1969).

Computation of the expected value of y requires computation of the following multi-dimensional integral over the restricted area defined by the selection. Let $\Omega = \{x_i \geq u_i; 1 \leq i \leq k\}$, $U = \{u_1, \dots, u_k\}$, ϕ_k : the density function of the multivariate normal distribution. The integral is given by $E_{\Omega}(y) = \int_{-\infty}^{\infty} y(x) \int_0^{\infty} \dots \int_0^{\infty} \phi_k(X; U, \Sigma) dx_1 \dots dx_k dy$

For further details about the whole project (selection gain for Double haploids and selection gain with markers), please contact the project contact person. The most important function is "multistageselection", which calculates the selection gain from given selection fraction and correlation matrix.

Value

The value returned, is the gain of selection.

Note

No further notes

Author(s)

Xuefei Mi, H. Friedrich Utz, Albrecht E. Melchinger

Maintainer: Xuefei Mi, <mi at pz.uni-hohenheim.de>

Contact person: Albrecht E. Melchinger, melchinger at uni-hohenheim.de

References

W.G. Cochran. Improvent by means of selection. In: Proceedings Second Berkeley Symposium on Math Stat Prof, pp449-470, 1951

G.M. Tallis. Moment generating function of truncated multi-normal distribution. Journal of the Royal Statistical Society, Series B, 23(1):223-229, 1961.

H.F. Utz. Mehrstufenselektion in der Pflanzenzuechtung. Doctor thesis, University Hohenheim, 1969.

X. Mi, T. Miwa and T. Hothorn. Implement of Miwa's analytical algorithm of multi-normal distribution, R Journal, 1:37-39, 2009.

A., Genz, F., Bretz. Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics, Vol. 195, Springer-Verlag, Heidelberg, 2009

A., Genz, F., Bretz, T., Miwa, X., Mi, F., Leisch, F., Scheipl, T., Hothorn. mvtnorm: Multivariate normal and t distributions. R package version 0.9-9, 2010.

See Also

No link

Examples

```
##---- Should be DIRECTLY executable !! ----  
##-- ==> Define data, use random,  
##--or do help(data=index) for the standard data sets.
```

```
corr=diag(4)
```

```
corr12=0.3508  
corr[1,2]=corr12  
corr[2,1]=corr12
```

```
corr13=0.3508  
corr[1,3]=corr13  
corr[3,1]=corr13
```

```
corr14=0.4979  
corr[1,4]=corr14  
corr[4,1]=corr14
```

```

corr23=0.3016
corr[2,3]=corr23
corr[3,2]=corr23

corr24=0.5630
corr[2,4]=corr24
corr[4,2]=corr24

corr34=0.5630
corr[3,4]=corr34
corr[4,3]=corr34

quantile=c(0.4308,0.9804,1.8603)
k=c(-200,quantile)
# -200 is small enough to send the lower bound to be -Inf

alphaofx=pmvnorm(lower=c(-200,quantile),corr=corr)

multistagegain.each(k=c(-200,quantile),corr=corr,alphaofx,sum.dim=4)

#####
# new code for adapt the window input of the red-R
#####

k=c(-200,0.4308,0.9804,1.8603)

corr=matrix( c(1,      0.3508,0.3508,0.4979,
               0.3508 ,1,      0.3016,0.5630,
               0.3508, 0.3016,1      ,0.5630,
               0.4979, 0.5630,0.5630,1),
             nrow=4
           )

sum.dim=4

alphaofx=pmvnorm(lower=k,corr=corr)

multistagegain.each(k=k,corr=corr,alphaofx,sum.dim=4)

# further examples 3 for the paper

alpha1<- 1/24
alpha2<- 1
Q=calculatefromalpha(alpha=c(alpha1,alpha2),dim=2,corr=corr[2:3,2:3])

k=c(-200,Q)

```

```
corr=matrix( c(1,      0.7071068, 0.9354143,
               0.7071068, 1,      0.7559289,
               0.9354143, 0.7559289, 1
             ),
             nrow=3
           )

sum.dim=3

alphaofx=pmvnorm(lower=k,corr=corr)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3,stages=TRUE)

multistagegain.each(k=k,corr=corr,alphaofx,sum.dim=3)

multistagevariance(k=k,corr=corr,alphaofx,sum.dim=3,alg=Miwa)

# further examples 4 for the paper

alpha1<- 1/(24)^0.5
alpha2<- 1/(24)^0.5
Q=calculatefromalpha(alpha=c(alpha1,alpha2),dim=2,corr=corr[2:3,2:3])

k=c(-200,Q)

corr=matrix( c(1,      0.7071068, 0.9354143,
               0.7071068, 1,      0.7559289,
               0.9354143, 0.7559289, 1
             ),
             nrow=3
           )

sum.dim=3

alphaofx=pmvnorm(lower=k,corr=corr)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3,stages=TRUE)

multistagegain.each(k=k,corr=corr,alphaofx,sum.dim=3)

multistagevariance(k=k,corr=corr,alphaofx,sum.dim=3,alg=Miwa)
```

multistageselection *Expected gain for k-stages selection (old function)*

Description

This function calculates the well-known selection gain $\Delta G(y)$, which is described by Cochran (1951). For one-stage selection the gain is defined as $\Delta G(y) = i\rho_y\rho_1$, where i is the selection intensity, ρ_1 is the correlation between the true breeding value and the selection index y (Utz 1969).

During a selection procedure, questions such as "How great is the yield after selection?" and "After three years testing and selection, how much is the chance that we miss one good candidate or the best candidate?" are frequently asked. These two questions can be answered by calculating the expected gain and variance of the gain, of a truncated multi-normal distribution (Tallis 1961). A numerical equation for calculating such moment generating function is since a long time available (Utz 1969). However, the properties of this old algorithm is limited. It is slow and can only handle a restricted selection model, which has at most three independent variables.

We developed an R-package for maximizing the gain of a multi-stage selection procedure under certain restrictions, e.g., a given annual budget or certain risk limits of each stage. This package is applied in the fields of plant/animal breeding, where a multi-normal regression model is commonly built. It can handle a restricted selection model with up to eight independent variables within seconds.

This is an old function, for R version ≤ 2.10 , it is replaced by multistagegain afterwards.

Usage

```
multistageselection(k, corr, alphaofx, sumdimofxandy,alg)
```

Arguments

sumdimofxandy	It is the dimension of k, which is dimension of x plus 1 (y has one dimension).
k	The lower bound of the integral, should be a vector. The first element of k is the lower bound of the response variable, Y, which is usually set as -Inf. The dimension of k should be the dimension of selection stage plus one.
corr	correlation matrix.
alphaofx	Selected fraction, this is a scalar not a vector.
alg	It decides which algorithm will be used, the Genz and Brett' algorithm is used by default. The Miwa's algorithm can be optional (Mi. et al. 2009; Genz. et al. 2010).

Details

There are only three functions in the package. The functions are used to calculate the moment generating function of the truncated multivariate normal distribution. The selection gain can be described as follow:

$x_i = y + e_i$, where x_i : observed mean value of the character at the i th stage, y : the true genetic value which generates the observation, e_i : experiment error, which is assumed to be

normally distributed. The true genetic value can be estimated by the observations with a regression function, $y(x)=f(x_1,x_2,\dots,x_i)$. A fraction u is defined as the ratio of the variance of e_1 and y

Under the normality assumption, y and x_i are multivariate normally distributed. We want to maximize the expected value of y in the restricted area, which is specified by the selection fraction. The Moment Generating Function (MGF) is used to calculate this multi-dimensional integral (Tallis, 1961; Utz, 1969).

Computation of the expected value of y requires computation of the following multi-dimensional integral over the restricted area defined by the selection. Let $\Omega = \{x_i \geq u_i; 1 \leq i \leq k\}$, $U = \{u_1, \dots, u_k\}$, ϕ_k : the density function of the multivariate normal distribution. The integral is given by $E_{\Omega}(y) = \int_{-\infty}^{\infty} y(x) \int_0^{\infty} \dots \int_0^{\infty} \phi_k(X; U, \Sigma) dx_1 \dots dx_k dy$

For further details about the whole project (selection gain for Double haploids and selection gain with markers), please contact the project contact person. The most important function is "multistageselection", which calculates the selection gain from given selection fraction and correlation matrix.

Value

The value returned, is the gain of selection.

Note

No further notes

Author(s)

Xuefei Mi, H. Friedrich Utz, Albrecht E. Melchinger

Maintainer: Xuefei Mi, <mi at pz.uni-hohenheim.de>

Contact person: Albrecht E. Melchinger, melchinger at uni-hohenheim.de

References

W.G. Cochran. Improvement by means of selection. In: Proceedings Second Berkeley Symposium on Math Stat Prof, pp449-470, 1951

G.M. Tallis. Moment generating function of truncated multi-normal distribution. Journal of the Royal Statistical Society, Series B, 23(1):223-229, 1961.

H.F. Utz. Mehrstufenselektion in der Pflanzenzuechtung. Doctor thesis, University Hohenheim, 1969.

X. Mi, T. Miwa and T. Hothorn. Implement of Miwa's analytical algorithm of multi-normal distribution, R Journal, 1:37-39, 2009.

A., Genz, F., Bretz. Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics, Vol. 195, Springer-Verlag, Heidelberg, 2009

A., Genz, F., Bretz, T., Miwa, X., Mi, F., Leisch, F., Scheipl, T., Hothorn. mvtnorm: Multivariate normal and t distributions. R package version 0.9-9, 2010.

See Also

No link

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

corr=diag(4)

corr12=0.3508
corr[1,2]=corr12
corr[2,1]=corr12

corr13=0.3508
corr[1,3]=corr13
corr[3,1]=corr13

corr14=0.4979
corr[1,4]=corr14
corr[4,1]=corr14

corr23=0.3016
corr[2,3]=corr23
corr[3,2]=corr23

corr24=0.5630
corr[2,4]=corr24
corr[4,2]=corr24

corr34=0.5630
corr[3,4]=corr34
corr[4,3]=corr34

quantile=c(0.4308,0.9804,1.8603)
k=c(-200,quantile)
# -200 is small enough to send the lower bound to be -Inf

alphaofx=pmvnorm(lower=c(-200,quantile),corr=corr)

multistageselection(k=c(-200,quantile),corr=corr,alphaofx,sumdimofxandy=4)

#####
# new code for adapt the window input of the red-R
#####

k=c(-200,0.4308,0.9804,1.8603)
```

```

corr=matrix( c(1,      0.3508,0.3508,0.4979,
              0.3508 ,1,      0.3016,0.5630,
              0.3508, 0.3016,1      ,0.5630,
              0.4979, 0.5630,0.5630,1),
            nrow=4
          )

sumdimofxandy=4

alphaofx=pmvnorm(lower=k,corr=corr)

multistageselection(k=k,corr=corr,alphaofx,sumdimofxandy=4)

```

multistagevariance *Expected gain for k-stages selection*

Description

During a selection procedure, questions such as "How great is the yield after selection?" and "After three years testing and selection, how much is the chance that we miss one good candidate or the best candidate?" are frequently asked. These two questions can be answered by calculating the expected gain and variance of the gain, of a truncated multi-normal distribution (Tallis 1961). A numerical algorithm for calculating such moment generating function is since a long time available (Utz 1969). However, the properties of this old algorithm is limited. It is slow and can only handle a restricted selection model, which has at most three independent variables.

We developed an R-package for maximizing the gain of a multi-stage selection procedure under certain restrictions, e.g., a given annual budget or certain risk limits of each stage. This package is applied in the fields of plant/animal breeding, where a multi-normal regression model is commonly built. It can handle a restricted selection model with up to eight independent variables within seconds.

Usage

```
multistagevariance(k, corr, alphaofx, sum.dim,alg)
```

Arguments

sum.dim	It is the dimension of k, which is dimension of x plus 1 (y has one dimension).
k	The lower bound of the integral, should be a vector. The first element of k is the lower bound of the response variable, Y, which is usually set as -Inf. The dimension of k should be the dimension of selection stage plus one.
corr	correlation matrix.
alphaofx	Selected fraction, this is a scalar not a vector.
alg	It decides which algorithm will be used, the Genz and Brett' algorithm is used by default. The Miwa's algorithm can be optional (Mi. et al. 2009; Genz. et al. 2010).

Details

There are only two functions in the package. The functions are used to calculate the moment generating function of the truncated multivariate normal distribution. The selection gain can be described as follow:

$x_i = y + e_i$, where x_i : observed mean value of the character at the i th stage, y : the true genetic value which generates the observation, e_i : experiment error, which is assumed to be normally distributed. The true genetic value can be estimated by the observations with a regression function, $y(x) = f(x_1, x_2, \dots, x_i)$. A fraction u is defined as the ratio of the variance of e_1 and y

Under the normality assumption, y and x_i are multivariate normally distributed. We want to maximize the expected value of y in the restricted area, which is specified by the selection fraction. The Moment Generating Function (MGF) is used to calculate this multi-dimensional integral (Tallis, 1961; Utz, 1969).

Computation of the expected value of y requires computation of the following multi-dimensional integral over the restricted area defined by the selection. Let $\Omega = \{x_i \geq u_i; 1 \leq i \leq k\}$, $U = \{u_1, \dots, u_k\}$, ϕ_k : the density function of the multivariate normal distribution. The integral is given by $E_{\Omega}(y) = \int_{-\infty}^{\infty} y(x) \int_0^{\infty} \dots \int_0^{\infty} \phi_k(X; U, \Sigma) dx_1 \dots dx_k dy$

For further details about the whole project (selection gain for Double haploids and selection gain with markers), please contact the project contact person. The most important function is "multistageselection", which calculates the selection gain from given selection fraction and correlation matrix.

Value

The output of this function is a vector that contains four values, which are the $E(y^2)$, $\sum_{k=1}^n B_k$, $\sum_{k=1}^n (\rho_{i,k} \sum_{r \neq k} C_{r,k})$ and α . The user have to notice that if they want the variance, they have to use $E(y^2)$ subtract the $[E(y)]^2$ got from last function.

If `sum.dim = 2`, the user have to be noticed. The value returned, is only the $E(y^2)$.

Note

No further notes

Author(s)

Xuefei Mi, H. Friedrich Utz, Albrecht E. Melchinger

Maintainer: Xuefei Mi, <mi at pz.uni-hohenheim.de>

Contact person: Albrecht E. Melchinger, melchinger at uni-hohenheim.de

References

W.G. Cochran. Improvement by means of selection. In: Proceedings Second Berkeley Symposium on Math Stat Prof, pp449-470, 1951

G.M. Tallis. Moment generating function of truncated multi-normal distribution. Journal of the Royal Statistical Society, Series B, 23(1):223-229, 1961.

H.F. Utz. Mehrstufenselektion in der Pflanzenzuechtung. Doctor thesis, University Hohenheim, 1969.

X. Mi, T. Miwa and T. Hothorn. Implement of Miwa's analytical algorithm of multi-normal distribution, R Journal, 1:37-39, 2009.

A., Genz, F., Bretz. Computation of Multivariate Normal and t Probabilities. Lecture Notes in Statistics, Vol. 195, Springer-Verlag, Heidelberg, 2009

A., Genz, F., Bretz, T., Miwa, X., Mi, F., Leisch, F., Scheipl, T., Hothorn. mvtnorm: Multivariate normal and t distributions. R package version 0.9-9, 2010.

See Also

No link

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

corr=diag(4)

corr12=0.3508
corr[1,2]=corr12
corr[2,1]=corr12

corr13=0.3508
corr[1,3]=corr13
corr[3,1]=corr13

corr14=0.4979
corr[1,4]=corr14
corr[4,1]=corr14

corr23=0.3016
corr[2,3]=corr23
corr[3,2]=corr23

corr24=0.5630
corr[2,4]=corr24
corr[4,2]=corr24

corr34=0.5630
corr[3,4]=corr34
corr[4,3]=corr34

quantile=c(0.4308,0.9804,1.8603)
k=c(-200,quantile)
# -200 is small enough to send the lower bound to be -Inf
```

```

alphaofx=pmvnorm(lower=c(-200,quantile),corr=corr)

multistageselection(k=c(-200,quantile),corr=corr,alphaofx,sumdimofxandy=4)

#####
# new code for adapt the window input of the red-R
#####

k=c(-200,0.4308,0.9804,1.8603)

corr=matrix( c(1,      0.3508,0.3508,0.4979,
               0.3508 ,1,      0.3016,0.5630,
               0.3508, 0.3016,1      ,0.5630,
               0.4979, 0.5630,0.5630,1),
             nrow=4
           )

sumdimofxandy=4

alphaofx=pmvnorm(lower=k,corr=corr)

multistageselection(k=k,corr=corr,alphaofx,sumdimofxandy=4)

selection.var.time.miwa=system.time (var.miwa<-multistagevariance(k=k,corr=corr,alphaofx,sum.dim=4,alg=Miwa))

selection.var.time.bretz=system.time (var.bretz<-multistagevariance(k=k,corr=corr,alphaofx,sum.dim=4))

selection.var.time.miwa
var.miwa[1]
selection.var.time.bretz
var.bretz[1]

# further examples 1

k=c(-200,0.9674216, 1.6185430)

corr=matrix( c(1,      0.7071068, 0.9354143,
               0.7071068, 1,      0.7559289,
               0.9354143, 0.7559289, 1
               ),
             nrow=3
           )

sum.dim=3

alphaofx=pmvnorm(lower=k,corr=corr)

```

```

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3,stages=TRUE)

multistagevariance(k=k,corr=corr,alphaofx,sum.dim=3,alg=Miwa)

selection.var.time.miwa=system.time (var.miwa<-multistagevariance(k=c(-200,quantile),corr=corr,alphaofx,sum.dim=3,alg=Miwa))

selection.var.time.bretz=system.time (var.bretz<-multistagevariance(k=c(-200,quantile),corr=corr,alphaofx,sum.dim=3,alg=Bretz))

selection.var.time.miwa
var.miwa[1]
selection.var.time.bretz
var.bretz[1]

# further examples 2

alpha1<- 1
alpha2<- 1/24
Q=calculatefromalpha(alpha=c(alpha1,alpha2),dim=2,corr=corr[2:3,2:3])

k=c(-200,Q)

corr=matrix( c(1,      0.7071068, 0.9354143,
               0.7071068, 1,      0.7559289,
               0.9354143, 0.7559289, 1
             ),
             nrow=3
           )

sum.dim=3

alphaofx=pmvnorm(lower=k,corr=corr)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3,stages=TRUE)

multistagevariance(k=k,corr=corr,alphaofx,sum.dim=3,alg=Miwa)

# further examples 3 for the paper

alpha1<- 1/6
alpha2<- 1/4
Q=calculatefromalpha(alpha=c(alpha1,alpha2),dim=2,corr=corr[2:3,2:3])

k=c(-200,Q)

```

```

corr=matrix( c(1,      0.7071068, 0.9354143,
               0.7071068, 1,      0.7559289,
               0.9354143, 0.7559289, 1
             ),
             nrow=3
)

sum.dim=3

alphaofx=pmvnorm(lower=k,corr=corr)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3,stages=TRUE)

multistagegain.each(k=k,corr=corr,alphaofx,sum.dim=3)

multistagevariance(k=k,corr=corr,alphaofx,sum.dim=3,alg=Miwa)

# further examples 4 for the paper

alpha1<- 1/(24)^0.5
alpha2<- 1/(24)^0.5
Q=calculatefromalpha(alpha=c(alpha1,alpha2),dim=2,corr=corr[2:3,2:3])

k=c(-200,Q)

corr=matrix( c(1,      0.7071068, 0.9354143,
               0.7071068, 1,      0.7559289,
               0.9354143, 0.7559289, 1
             ),
             nrow=3
)

sum.dim=3

alphaofx=pmvnorm(lower=k,corr=corr)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3)

multistagegain(k=k,corr=corr,alphaofx,sum.dim=3,stages=TRUE)

multistagegain.each(k=k,corr=corr,alphaofx,sum.dim=3)

multistagevariance(k=k,corr=corr,alphaofx,sum.dim=3,alg=Miwa)

```

Index

*Topic **Non-equi coordinate quantile
(NCQ)**

calculatefromalpha, [2](#)

*Topic **Truncated multivariate
normal**

multistagegain, [4](#)

multistagegain.each, [7](#)

multistageselection, [12](#)

multistagevariance, [15](#)

calculatefromalpha, [2](#)

multistagegain, [4](#)

multistagegain.each, [7](#)

multistageselection, [12](#)

multistagevariance, [15](#)