Package 'VarRedOpt'

October 12, 2022

Type Package

Title A Framework for Variance Reduction

Version 0.1.0

Description In order to make it easy to use variance reduction algorithms for any simulation, this framework can help you.
We propose user friendly and easy to extend framework. Antithetic Variates, Inner Control Variates,
Outer Control Variates and Importance Sampling algorithms are available in the framework. User can
write its own simulation function and use the Variance Reduction techniques in this package to obtain more efficient simulations. An implementation of Asian Option simulation is already available within the package. See Kemal Dinçer Dingeç & Wolfgang Hörmann (2012) <doi:10.1016/j.ejor.2012.03.046>.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests testthat

NeedsCompilation no

Author Onur Boyar [aut, cre],
Wolfgang Hörmann [aut]

Maintainer Onur Boyar <boyaronur@gmail.com>

Repository CRAN

Date/Publication 2020-12-08 10:30:07 UTC

R topics documented:

`BS_Asian_geom` *Block Scholes for Geometric Asian Call Option*

Description

Function to calculate expected value of Geometric Asian Call Option via Block Scholes formula.

Usage

```
BS_Asian_geom(K = 100, TimeToMat, d, ti, r = 0.05, sigma = 0.1, S0 = 100, ...)
```

Arguments

`K`	Strike price.
`TimeToMat`	Time to maturity.
`d`	Dimension of input z matrix.
`ti`	Vector of control points.
`r`	Riskfree rate
`sigma`	Yearly volatility.
`S0`	Stock price at start.
`...`	ellipsis parameter. different parameters can be passed depending on the problem.

Value

Expected value of Geometric Average Asian Call Option, vector of control points, interest rate and strike price as a list.

Examples

```
sim.outer(n=1e3, d=3, q.outer = myq_asian, K=100, ti=(1:3/12), r=0.03, sigma=0.3, S0=100)
```

myq_asian *Block Scholes for Geometric Asian Call Option*

Description

Function to calculate expected value of Geometric Asian Call Option via Block Scholes formula.

Usage

```
myq_asian(zm, K = 100, ti = (1:3)/12, r = 0.05, sigma = 0.1, S0 = 100)
```

Arguments

zm	Input matrix with n row and d dimension.
K	Strike price.
ti	Vector of control points.
r	Riskfree rate.
sigma	Yearly volatility.
S0	Stock price at start.

Value

Returns 4 elements as a list. Asian Call Option Prices, Last Price of Asian Call Option, Expected Value of Asian Call Option, Product of the prices through time

Examples

```
 sim.outer(n=1e3, d=3, q.outer = sim.IS,
q.is = myq_asian, K=100, ti=(1:3/12), r=0.03, sigma=0.3, S0=100)
```

myq_euclidean *Euclidean Distance*

Description

Function to calculate euclidean distance between two vectors.

Usage

```
myq_euclidean(zm, point = c(1, 2, 1))
```

Arguments

zm	Input matrix of set of vectors.
point	Coordinates of the point to calculate distance to the input.

Value

Sum of the euclidean distance from point to set of vectors.

Examples

```
sim.outer(n=1e3, d=3, q.outer=myq_euclidean, point=c(1,1,1))
```

`sim.AV` *Function to apply Antithetic Variates Algorithm.*

Description

Given matrix input with d dimension, this function runs simulation two times using positive and negative signed versions of the input matrix.

Usage

```
sim.AV(zm, q.av, ...)
```

Arguments

`zm`	A matrix with dimension d and length n.
`q.av`	q function that sim.AV function gets target vectors to apply variance reduction.
`...`	ellipsis parameter. different parameters can be passed depending on the problem.

Value

y target vector with theoretically lower variance with the same expected value as the initial y vector.

Examples

```
 sim.outer(n=1e3, d=3, q.outer = sim.AV,
q.av = myq_asian, K=100, ti=(1:3/12), r=0.03, sigma=0.3, S0=100)

sim.outer(n=1e3, d=3, q.outer = sim.AV, q.av = sim.InnerCV,
q.cv = myq_asian, K=100, ti=(1:3/12), r=0.03, sigma=0.3, S0=100)
```

sim.GeometricAvg *An Outer Control Variate function for Asian Call Option.*

Description

Applies geometric average asian call outer control varites algorithm to the simulation. Gets expected value for the control variate using BS_Asian_geom function if IS algorithm is within the framework, the length of the q.ga will be different. Checks if IS algorithm is within the framework and applies IS weight accordingly.

Usage

```
sim.GeometricAvg(zm, q.ga, ...)
```

Arguments

zm	A matrix with dimension d and length n.
q.ga	q function that sim.GeometricAvg function gets target vectors to apply variance reduction.
...	ellipsis parameter. different parameters can be passed depending on the problem.

Value

Updates Y value which stored in list 'results' and returns the list 'results' with updated Y value.

Examples

```
 sim.outer(n=1e3, d=3, q.outer = sim.GeometricAvg,
q.ga = myq_asian, K=100, ti=(1:3/12), r=0.03, sigma=0.3, S0=100)

sim.outer(n=1e3, d=1, q.outer = sim.AV, q.av = sim.GeometricAvg,
q.ga = myq_asian, K=90, ti=(1:1/12), r=0.03, sigma=0.3, S0=100)
```

sim.InnerCV *Function to apply Inner Control Variates Algorithm.*

Description

Given matrix input with d dimension, this function aims to reduce the variance by applying Inner Control Variates algorithm. It uses input columns and their squares as inner control variates and applies feature selection for these control variates.

Usage

```
sim.InnerCV(zm, q.cv, ...)
```

Arguments

zm	A matrix with dimension d and length n.
q.cv	q function that sim.InnerCV function gets target vectors to apply variance reduction.
...	ellipsis parameter. different parameters can be passed depending on the problem.

Value

Updates Y value which stored in list 'results' and returns the list 'results' with updated Y value.

Examples

```
 sim.outer(n=1e3, d=3, q.outer = sim.InnerCV,
q.cv = myq_asian, K=100, ti=(1:3/12), r=0.03, sigma=0.3, S0=100)

sim.outer(n=1e6, d=6, q.outer = sim.AV, q.av = sim.InnerCV,
q.cv = myq_asian, K=105, ti=(1:6/12), r=0.03, sigma=0.3, S0=100)
```

---

sim.IS *Function to apply Importance Sampling Algorithm.*

Description

Given matrix input with d dimension, this function applies Importance Sampling algorithm and it chooses the best value of the mean value of the importance density automatically. Performs better in rare event simulation. sim.IS checks the input size and returns elements accordingly. If the input value is a list of one element it only updates the simulated value with importance weight and returns that value. If the input value has a dimension bigger than 1, weight values are added separately as another input of the list.

Usage

```
sim.IS(zm, use_pilot_study = TRUE, muis = 1, sis = 1, q.is, ...)
```

Arguments

zm	A matrix with dimension d and length n.
use_pilot_study	TRUE if user wants to choose muis parameter automatically.
muis	mean parameter of the importance density.
sis	standard deviation parameter of the importance density.

q.is	q function that sim.IS function gets target vectors to apply variance reduction.
...	ellipsis parameter. different parameters can be passed depending on the problem.

Value

Weighted simulated values or weights and simulated values are added to input list depending on the initial input size.

Examples

```
 sim.outer(n=1e3, d=3, q.outer = sim.IS,
q.is = myq_asian, K=100, ti=(1:3/12), r=0.03, sigma=0.3, S0=100)

sim.outer(n=1e5, d=4, q.outer=sim.AV, q.av = sim.IS, q.is = myq_asian,
         K=130, ti=(1:4/12), r=0.03, sigma=0.3, S0=100)
```

---

sim.outer *Main function for VarRedOpt simulation framework.*

Description

This function creates the z matrix which is an input matrix with given dimension value, d, and given length, n. Values are generated from standard normal distribution. After creating the z matrix, this function sends this input matrix to given simulation function. After simulation steps are completed, simulate.outer function gets the final simulated values and calculates expected value and variance. For instance, if myq_asian and sim.AV functions are given in simulate.outer function as parameters, the input matrix will be sent to sim.AV function and sim.AV function will send input value to myq_asian function twice with opposite signs and gets simulation results. After collecting these results it applies Antithetic Variates algorithm and finds the final simulation value and sends it back to the simulate.outer function.

Usage

```
sim.outer(n, d, auto_repetition = 1, q.outer, ...)
```

Arguments

n	Simulation length.
d	Simulation dimension.
auto_repetition	Applies auto_repetition of auto_repetition = TRUE.
q.outer	Accepts the function name of the variance reduction / simulation algorithm.
...	ellipsis parameter. different parameters can be passed depending on the problem.

Value

estimation mean, standard error, confidence interval metrics if auto_repetition = TRUE

Examples

```
 sim.outer(n=1e3, d=3, q.outer = sim.AV,
q.av = myq_asian, K=100, ti=(1:3/12), r=0.03, sigma=0.3, S0=100)
```

Index