Numerics in Applied Mathematical Finance (with R)

Maxim Litvak

2017-03-06

Outline

Numerical derivative - I

2 point symmetric derivative

$f'_{num}(x) = \frac{f(x+h)-f(x-h)}{2h}$

Both advantage and disadvantage of symmetry:

More derivatives exist (e.g. modulus function), but some of them you may not want to exist

Question: how to set h?

Set h depending on application cases.

Numerical derivative - II

Code example

```
num.deriv <- function(f, x, h = 1e-05)
{
  return((f(x + h) - f(x - h))/(2*h))
}

print(num.deriv(sqrt, 4, .1))
print(num.deriv(sqrt, 4, .01))

[1] 0.2500195
[1] 0.2500002
```

Numerical derivative - III

5 point derivative

If 2-point derivative behaves badly, try more precision

$f'_{num} = \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12h}$

Semidefinite matrix decomposition and eigenvalues

Example

Assume that the default rates in different industries are correlated. The corresponding correlation matrix is positive semidefinite.

In order to draw the correlated random variables we need to decompose the correlation matrix

Possibilities

Cholesky decomposition

$\Sigma = LL^T$ where L is low-triangular

Eigendecomposition

$\Sigma = (Q\Lambda^{\frac{1}{2}})(Q\Lambda^{\frac{1}{2}})^T$ where Λ is matrix with eigenvalues on diagonal and 0 elsewhere, Q is matrix of eigenvectors

Pseudo-Random Numbers Generators (PRNG) - I

Importance of reproducibility of results

Sounds like a paradox: random numbers must be reproducible

Helps to spot the effects of other factors, e.g. you need to calculate impact of the new pricing algorithm and eliminate effect of randomness.

Setting seeds

- "Whatever one sows, that will he also reap"
- Input: one number (called "seed"), output: the sequence of numbers that repeats only after very big period
- Period of currently popular Mersenne Twister is $2

^19937 - 1$

- RAND() function in Excel2003 has period of $10^1 3$
- However, there are still legacy systems in use with small period (e.g. 40 mln)

Pseudo-Random Numbers Generators (PRNG) - II

Code example

```
loss.dist <- function(seed, N)
{
  set.seed(seed)
  return(runif(N))
}

print(loss.dist(seed=1, N=4))
print(loss.dist(seed=1, N=4))  # same seed - same sequence
print(loss.dist(seed=10, N=4)) # different seed - different se

[1] 0.2655087 0.3721239 0.5728534 0.9082078
[1] 0.2655087 0.3721239 0.5728534 0.9082078
[1] 0.5074782 0.3067685 0.4269077 0.6931021
```

Correlated numbers generation

Practical scenario:

There's a correlation matrix given. However, an expert sets some of the negative correlations to 0 (reality check). We need to know if the adjusted matrix is still positive semidefinite.

Approach

The smallest eigenvalue must be positive.

Code example

```
R <- matrix(c(1,.5,.5,1))
print(min(eigen(R)$value))

        [,1]    [,2]
[1,] 1.00000 0.49559
[2,] 0.49559 1.00000
```

Correlated numbers generation - II

Code example

```
R <- matrix(c(1,.5,.5,1), nrow = 2)
EG <- eigen(R)
mx <- EG$vectors %*% diag(sqrt(EG$values))
V <- matrix(rnorm(1000), nrow = 2)
print(cor(t(mx%*%V)))
```

Computation of quantile functions - I

Given

F() - cdf, probability y Find: quantile x, s.t. y = F(x)

No closed form solution examples

- Normal distribution (not even cdf is given in elementary functions!)
- Gamma distribution

Quantile function given

If the quantile function is given, it's better to used its Taylor expention

Example

Normal cdf is implemented in practice as a piecwise Taylor polynom, i.e. with coefficients varying on different intervals.

Computation of quantile functions - II

Problem

Many algorithms require an interval to be defined, however, the quantile function are often defined on unconstrained intervals.

Example

- Find a quantile for Gamma distribution
- Problem: the right end of domain is unconstrained, the root finding algorithm doesn't converge in the tail
- Solution: use Chebyshev's inequality to constrain the domain

Computation of quantile functions - III

Application

Inequality

$P(|X - \mu| \geq 10\sigma) = 0.01$

In numbers

- Default rate 2%, $\theta = 1$
- $P(|X - 0.02| \geq 10 \times 0.02) = 0.01$
- Thus, right bound 0.22. If x is bigger than 0.22, then set it hard to 0.99 (if precision in the tail is not important)

Inequality application

$P(|X - \mu| \geq 10\sigma) = 0.01$

Computation of quantile functions - IV

Problem

Find quantile of Gamma distribution using uniroot procedure

Solution

```
pg <- function(x) pgamma(x, 0.02, 1) - 0.95
uniroot(pg, c(0,0.22))$root
# check: qgamma(0.95, 0.02, 1)

[1] 0.04592691
```