R Tools for Portfolio Optimization

Guy Yollin
Quantitative Research Analyst
Rotella Capital Management
Bellevue, Washington
Backgrounder

- Rotella Capital Management
  - Quantitative Research Analyst
    - Systematic CTA hedge fund trading 80+ global futures and foreign exchange markets

- Insightful Corporation
  - Director of Financial Engineering
    - Developers of S-PLUS®, S+FinMetrics®, and S+NuOPT®

- J.E. Moody, LLC
  - Financial Engineer
    - Futures Trading, Risk Management, Business Development

- OGI School of Engineering at Oregon Health & Science University
  - Adjunct Instructor
    - Statistical Computing & Financial Time Series Analysis

- Electro Scientific Industries, Inc
  - Director of Engineering, Vision Products Division
    - Machine Vision and Pattern Recognition

- Started Using R in 1999
R-SIG-FINANCE QUESTION:
Can I do <fill in the blank> portfolio optimization in R?

ANSWER:
Yes! (98% confidence level)
Outline

• Mean-Variance Portfolio Optimization
  • quadratic programming
    • tseries, quadprog

• Conditional Value-at-Risk Optimization
  • linear programming
    • Rglpk_solve_LP package

• General Nonlinear Optimization
  • Differential Evolution Algorithm
    • DEoptim package
      • Omega Optimization
      • Adding Constraints
      • Maximum Drawdown Optimization
      • R-Ratio Optimization

• Wrap-Up
Efficient Portfolio Solution

DJIA Returns: 02/04/2009 - 04/03/2009

Portfolio Weights
Mean-Variance Portfolio Optimization

- Function
  - `portfolio.optim` \{tseries\}

- Description
  - computer mean-variance efficient portfolio

- Usage

  ```r
  portfolio.optim(x, pm = mean(x), riskless = FALSE, shorts = FALSE, 
  rf = 0.0, reslow = NULL, rehigh = NULL, covmat = cov(x), ...)
  ```

- Example

  ```r
  > averet = matrix(colMeans(r), nrow=1)
  > rcov = cov(r)
  > target.return = 15/250
  > port.sol = portfolio.optim(x = averet, pm = target.return, 
  covmat = rcov, shorts = F, reslow = rep(0,30), rehigh = rep(0.1,30))
  > w = cleanWeights(port.sol$pw,syms)
  > w[w!=0]
  
  HD  IBM  INTC  JNJ  KO  MCD  MSFT  PFE  PG  T  VZ  WMT
  0.05 0.10 0.04 0.10 0.10 0.10 0.07 0.04 0.10 0.10 0.10 0.10
  ```
Efficient Frontier

DJIA Returns: 02/04/2009 - 04/03/2009

efficient frontier with shorts

efficient frontier long only
Efficient Frontier Calculation

effFrontier = function (averet, rcov, nports = 20, shorts=T, wmax=1) {
    mxret = max(abs(averet))
mnret = -mxret
n.assets = ncol(averet)
reshigh = rep(wmax,n.assets)
if( shorts )
{
    reslow = rep(-wmax,n.assets)
} else {
    reslow = rep(0,n.assets)
}
min.rets = seq(mnret, mxret, len = nports)
vol = rep(NA, nports)
ret = rep(NA, nports)
for (k in 1:nports)
{
    port.sol = NULL
    try(port.sol <- portfolio.optim(x=averet, pm=min.rets[k], covmat=rcov,
                                   reshigh=reshigh, reslow=reslow, shorts=shorts),silent=T)
    if ( !is.null(port.sol) ) 
    {
        vol[k] = sqrt(as.vector(port.sol$pw %*% rcov %*% port.sol$pw))
        ret[k] = averet %*% port.sol$pw
    }
}
return(list(vol = vol, ret = ret))
}
Maximum Sharpe Ratio

DJIA Returns: 02/04/2009 - 04/03/2009

R Tools for Portfolio Optimization
Maximum Sharpe Ratio

```r
define maxSharpe function

```r
define optim.callback function

```r
define ef function

```r
define ncol function

```r
define reshig = rep function

```r
define if...else...

```r
define max.sh = which.max function

```r
define w = rep function

```r
define xmin = optimize function

```r
define return function

```
Solving Quadratic Programs

- **Function**
  - `solve.QP {quadprog}`

- **Description**
  - solve quadratic program

  general quadratic program
  
  Minimize: \(-d^T b + \frac{1}{2} b^T D b\)
  
  Subject to: \(A^T b \geq b_0\)

  mean-variance portfolio optimization
  
  Minimize: \(w^T \Omega w\)
  
  Subject to:
  
  \[ \sum_i \bar{r}_i w_i = r_{\text{min}} \]
  
  \[ \sum_i w_i = 1 \]
  
  \[ w_i^{\text{min}} \leq w_i \leq w_i^{\text{max}} \]

- **Usage**

  `solve.QP(Dmat, dvec, Amat, bvec, meq=0, factorized=FALSE)`
Extending portfolio.optim

- Modify portfolio.optim
  - Market neutral (weights sum to zero)

```r
if (!is.null(reslow) & !is.null(reshigh)) {
  a3 <- matrix(0, k, k)
  diag(a3) <- 1
  Amat <- t(rbind(a1, a2, a3, -a3))
  b0 <- c(weight.sum, pm, reslow, -reshigh)
} else {
  Amat <- t(rbind(a1, a2))
  b0 <- c(weight.sum, pm)
}
```

- Call solve.QP directory
  - add group constraints
  - add linear transaction cost constraints
  - etc.
Outline

• Mean-Variance Portfolio Optimization
  • quadratic programming
    • tseries, quadprog

• Conditional Value-at-Risk Optimization
  • linear programming
    • Rglpk_solve_LP package

• General Nonlinear Optimization
  • Differential Evolution Algorithm
    • DEoptim package
      • Omega Optimization
      • Adding Constraints
      • Maximum Drawdown Optimization
      • R-Ratio Optimization

• Wrap-Up
Conditional Value-at-Risk

\[ \beta \text{ is our confidence level (e.g. 95% or 90%)} \]

mean of worst \(1-\beta\) losses
CVaR Optimization as a Linear Program

- CVaR

\[
R_{CVAR}(w, \beta) = R_{VAR} + \frac{1}{S} \sum_{s=1}^{S} \max(R_{VAR} - w'R_s, 0) \times \frac{1 - \beta}{\text{probability of excess loss}} \times \begin{array}{c}
\text{average excess loss over all scenarios} \\
\text{conditional excess loss} \\
\text{average loss if loss occurs}
\end{array}
\]

CVaR Optimization

Minimize:
\[
R_{VAR} + \frac{1}{S} \sum_{s=1}^{S} d_s
\]

Subject to:
\[
d_s \geq R_{VAR} + w'R_s \\
d_s \geq 0 \\
w'\mathbf{R} \geq R_{\text{min}} \\
\sum_i w_i = 1
\]

see B. Sherer 2003

R Tools for Portfolio Optimization
Solving Linear Programs

- **Function**
  - `Rglpk_solve_LP {Rglpk}`

- **Description**
  - Solves linear and MILP programs (via GNU Linear Programming Kit)

**General Linear Program**

\[
\text{Minimize: } \quad c^T x \\
\text{Subject to: } \quad Ax \geq b_0
\]

**CVaR Portfolio Optimization**

\[
c^T = \begin{bmatrix} 0 & 0 & \cdots & 0 & -\frac{1}{(1-\beta)S} & -\frac{1}{(1-\beta)S} & \cdots & -\frac{1}{(1-\beta)S} & -1 \end{bmatrix}
\]

\[
x^T = [w_1 \ w_2 \ \cdots \ w_n \ d_1 \ d_2 \ \cdots \ d_S \ \text{RVaR}]
\]

\[
A = \begin{bmatrix}
1 & 1 & \cdots & 1 & 0 & \cdots & 0 & 0 \\
\bar{r}_1 & \bar{r}_2 & \cdots & \bar{r}_n & 0 & \cdots & 0 & 0 \\
r_{11} & r_{12} & \cdots & r_{1n} & 1 & 0 & \cdots & 1 \\
r_{21} & r_{22} & \cdots & r_{2n} & 0 & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & 1 \\
r_{s1} & r_{s2} & \cdots & r_{sn} & 0 & \cdots & 1 & 1
\end{bmatrix}
\]

\[
b_0 = \begin{bmatrix} 1 \\
r_{\text{min}} \\
0 \\
\vdots \\
0 \end{bmatrix}
\]

**Usage**

\[
\text{Rglpk_solve_LP(obj, mat, dir, rhs, types = NULL, max = FALSE, bounds = NULL, verbose = FALSE)}
\]
CVaR Optimization

cvarOpt = function(rmat, alpha=0.05, rmin=0, wmin=0, wmax=1, weight.sum=1) {
  require(Rglpk)
  n = ncol(rmat) # number of assets
  s = nrow(rmat) # number of scenarios i.e. periods
  averet = colMeans(rmat)

  # create objective vector, constraint matrix, constraint rhs
  Amat = rbind(cbind(rbind(1,averet),matrix(data=0,nrow=2,ncol=s+1)),
               cbind(rmat,diag(s),1))
  objL = c(rep(0,n), rep(-1/(alpha*s), s), -1)
  bvec = c(weight.sum, rmin, rep(0,s))

  # direction vector
  dir.vec = c("==",">=",rep(">",s))

  # bounds on weights
  bounds = list(lower = list(ind = 1:n, val = rep(wmin,n)),
                 upper = list(ind = 1:n, val = rep(wmax,n)))

  res = Rglpk_solve_LP(obj=objL, mat=Amat, dir=dir.vec, rhs=bvec,
                       types=rep("C",length(objL)), max=T, bounds=bounds)

  w = as.numeric(res$solution[1:n])
  return(list(w=w,status=res$status))
}

supports general equality/inequality constraints
CVaR Efficient Frontier

DJIA: 12/02/2008 - 04/15/2009

maximum CVaR ratio

minimum CVaR

Note: some assets not displayed due to cropping
Outline

• Mean-Variance Portfolio Optimization
  • quadratic programming
    • tseries, quadprog

• Conditional Value-at-Risk Optimization
  • linear programming
    • Rglpk_solve_LP package

• General Nonlinear Optimization
  • Differential Evolution Algorithm
    • DEoptim package
      • Omega Optimization
      • Adding Constraints
      • Maximum Drawdown Optimization
      • R-Ratio Optimization

• Wrap-Up
Differential Evolution

• DE is a very **simple** and yet very **powerful** population based stochastic function minimizer

• Ideal for global optimization of multidimensional multimodal functions (i.e. *really hard problems*)

• Developed in mid-1990 by Berkeley researchers Ken Price and Rainer Storm

• Implemented in R in the package DEoptim
Differential Evolution Algorithm

Permutation 1

<table>
<thead>
<tr>
<th>x1</th>
<th>0.32</th>
<th>0.29</th>
<th>0.68</th>
<th>-0.51</th>
<th>-1.29</th>
<th>-0.94</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>C</td>
<td>D</td>
<td>B</td>
<td></td>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

Permutation 2

<table>
<thead>
<tr>
<th>x1</th>
<th>1.29</th>
<th>-0.94</th>
<th>0.32</th>
<th>0.29</th>
<th>0.68</th>
<th>-0.51</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>A</td>
<td>E</td>
<td>C</td>
<td>D</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

Permutation 3

<table>
<thead>
<tr>
<th>x1</th>
<th>0.29</th>
<th>0.68</th>
<th>-0.51</th>
<th>1.29</th>
<th>-0.94</th>
<th>0.32</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>E</td>
<td>F</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

Mutant Population

<table>
<thead>
<tr>
<th>x1</th>
<th>-0.48</th>
<th>1.66</th>
<th>-0.23</th>
<th>0.65</th>
<th>-0.45</th>
<th>-0.02</th>
</tr>
</thead>
<tbody>
<tr>
<td>x2</td>
<td>1.31</td>
<td>-2.76</td>
<td>-0.88</td>
<td>-0.74</td>
<td>2.51</td>
<td>1.06</td>
</tr>
</tbody>
</table>

Random Crossover Template

<table>
<thead>
<tr>
<th>x1</th>
<th>F</th>
<th>F</th>
<th>T</th>
<th>T</th>
<th>F</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>x2</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

CR

Current Population

<table>
<thead>
<tr>
<th>x1</th>
<th>-0.94</th>
<th>-0.51</th>
<th>0.29</th>
<th>0.68</th>
<th>0.32</th>
<th>1.29</th>
</tr>
</thead>
<tbody>
<tr>
<td>x2</td>
<td>1.78</td>
<td>0.64</td>
<td>0.52</td>
<td>-1.75</td>
<td>0.15</td>
<td>-0.84</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
</table>

Next Gen Population

<table>
<thead>
<tr>
<th>x1</th>
<th>-0.94</th>
<th>-0.51</th>
<th>0.29</th>
<th>0.65</th>
<th>0.32</th>
<th>1.29</th>
</tr>
</thead>
<tbody>
<tr>
<td>x2</td>
<td>1.31</td>
<td>0.64</td>
<td>0.52</td>
<td>-0.74</td>
<td>0.15</td>
<td>1.06</td>
</tr>
</tbody>
</table>

see http://www.icsi.berkeley.edu/~storn/code.html

R Tools for Portfolio Optimization
DE Example

$$f(x) = (1 - x_1)^2 + 100(x_2 - x_1^2)^2$$

global minimum at (1,1)
**Differential Evolution Function**

- **Function**
  - `DEoptim` \{DEoptim\}

- **Description**
  - Performs evolutionary optimization via differential evolution algorithm

- **Usage**
  
  ```r
  DEoptim(FUN, lower, upper, control = list(), ...)
  ```

- **Example**

  ```r
  > lower = c(-2,-2)
  > upper = c(2,2)
  > res = DEoptim(banana, lower, upper)
  > res$optim
  $bestmem
    par1  par2
  0.9987438 0.9976079
  $bestval
  [1] 2.986743e-06
  $nfeval
  [1] 5050
  $iter
  [1] 100
  ```
Omega Performance Measure

Portfolio Returns Cumulative Distribution Function

\[ \Omega(L) = \frac{\int_{L}^{b} (1 - F(r)) \, dr}{\int_{a}^{L} F(r) \, dr} \]

Where:
- \( F(r) \) is the cumulative distribution function of daily returns.
- \( a \) and \( b \) are the lower and upper bounds of the distribution, respectively.
- \( L \) is the lower threshold for the Omega Measure.

The diagram illustrates the cumulative distribution function of daily returns, with the shaded areas representing the proportion of returns that are below or above certain thresholds.
Omega Performance Measure

- Omega Performance Measure:
  \[ \Omega(L) = \frac{\frac{1}{L} \int_a^b (1 - F(r)) \, dr}{\frac{1}{L} \int_a^b F(r) \, dr} \]

- utilizes entire returns distribution
- ratio of call price to put price with strike at L:
  \[ \Omega(L) = \frac{C(L)}{P(L)} \]
- simple calculation:
  \[ \text{omega} = \frac{\text{mean}(\text{pmax}(r-L, 0))}{\text{mean}(\text{pmax}(L-r, 0))} \]

See
Keating & Shadwick 2002
Kazemi et. al., 2003
Omega Optimization

Maximize: \( \Omega(L) \)
Subject to: \[ \sum_i |w_i| = 1 \]
\[ 0 \leq w_i \leq w_i^{\text{max}} \]

```
optOmega = function(x, ret, L)
{
  retu = ret %*% x
  obj = -Omega(retu, L=L, method="simple")
  weight.penalty = 100*(1-sum(x))^2
  return( obj + weight.penalty )
}
```

```
> lower = rep(0, n.assets)
> upper = rep(wmax, n.assets)

> res = DEoptim(optOmega, lower, upper,
  control=list(NP=2000, itermax=1000, F=0.2, CR=0.8),
  ret=coredata(r), L=L)

> w = cleanWeights(res$optim$bestmem, sym)
> w[w!=0]
          AXP      BA     C     CAT    CVX      DD     DIS    GE     GM     HD     IBM   INTC     KO    MCD    MMM
0.02 0.03 0.02 0.04 0.05 0.08 0.01 0.02 0.01 0.03 0.04 0.09 0.05 0.08 0.05 0.04
          MRK      PG     T   UTX     VZ    WMT   XOM
0.04 0.10 0.08 0.04 0.06 0.03 0.00
```
Effect of DE Parameters on Optimization

Optimal Omega as a function of F and CR

CR
NP=600, itermax=200

R Tools for Portfolio Optimization
Max Omega versus Max Sharpe

Max Omega
Omega = 3.50
Sharpe = 5.92

Max Sharpe
Sharpe = 6.76
Omega = 3.07

Note: some assets not displayed due to cropping
Weight Comparison

Weights of Max Sharpe Portfolio
Omega = 3.07

Weights of Max Omega Portfolio
Omega = 3.5
Optimization with Additional Constraints

```r
# max omega with non-zero weights between 3% & 10%
optOmega.gt3 = function(x, ret, L)
{
  retu = ret %*% x
  obj = -Omega(retu, L=L, method="simple")
  weight.penalty = 100*(1-sum(x))^2
  small.weight.penalty = 100*sum(x[x<0.03])
  return( obj + weight.penalty + small.weight.penalty )
}

res = DEoptim(optOmega.gt3, lower, upper, control=list(NP=2000, itermax=1000, F=0.2, CR=0.8), ret=coredata(r), L=L)
```

Weights of Max Omega Portfolio - non-zero weights 3%-10%

Omega = 2.88

all non-zero weights now greater than 3%
Maximum Drawdown

IBM: 12/02/2008 - 04/15/2009

Stock price

- Maximum Drawdown

IBM Underwater Graph
Maximum Drawdown Optimization

```r
# max drawdown with non-zero weights between 3% & 10%
optMDD.gt3 = function(x, ret)
{
  retu = ret %*% x
  obj = mddx(retu, 1)
  weight.penalty = 100*(1-sum(x))^2
  small.weight.penalty = 100*sum(x[x<0.03])
  return(obj + weight.penalty + small.weight.penalty)
}

res = DEoptim(optMDD.gt3, lower, upper,
  control=list(NP=2000, itermax=1000, F=0.2, CR=0.8),
  ret=coredata(r))
```

function return the mean of the top n drawdowns (in this case n=1)

could readily implement optimization on Calmar or Sterling ratios

Weights of Portfolio Optimized on Maximum Drawdown

R Tools for Portfolio Optimization
Maximum Drawdown Optimization

Optimised on Max Drawdown
MaxDD = 0.47%

Optimised on Omega Ratio
MaxDD = 1.85%

Optimised on Sharpe Ratio
MaxDD = 0.98%
Rachev Ratio (R-Ratio)

\[ R-Ratio = \frac{CVaR(\alpha)^+}{CVaR(\alpha)^-} \]

P/L Distribution

Daily return distribution with CVaR+ and CVaR-.
R-Ratio Optimization

```r
optRR.gt3 = function(x, ret)
{
  retu = ret %*% x
  obj = -CVaR(-retu)/CVaR(retu)
  weight.penalty = 100*(1-sum(x))^2
  small.weight.penalty = 100*sum(x[x<0.03])
  return( obj + weight.penalty + small.weight.penalty )
}

res = DEoptim(optRR.gt3, lower, upper, control=list(NP=2000, itermax=1000, F=0.2, CR=0.8), ret=coredata(r))
```

Weights of Portfolio Optimized on R-Ratio

![Weights of Portfolio Optimized on R-Ratio](image-url)
R-Ratio Comparison

Optimal R-Ratio

R-Ratio = 2.6

Optimal Max DD

R-Ratio = 1.7

Optimal Omega

R-Ratio = 1.9

Optimal Sharpe

R-Ratio = 2.1
Summary

• Mean-Variance Portfolio Optimization
  • high-level function: portfolio.optim
  • low-level function: solve.QP

• Linear Programming Optimization
  • Rglpk_solve_LP()
    • Conditional Value at Risk (CVaR):
    • MAD, Semivariance & others

• General-purpose non-linear optimization
  • DEoptim
    • Omega
    • non-linear constraints
    • maximum drawdown
    • R-Ratio
    • 130/30 portfolio optimization
Thank You

• Questions & Answers

• Contact the Author
  • Guy Yollin
  • guy.yollin@rotellacapital.com

• Learn more about Rotella Capital Management
  • businessdevelopment@rotellacapital.com
  • (425)-213-5700