Portfolio Selection: Recent Approaches
Optimization and Design with R

Bernhard Pfaff
bernhard_pfaff@fra.invesco.com

Invesco Asset Management Deutschland GmbH, Frankfurt am Main

7th R/Rmetrics Meielisalp Workshop
June 30 – July 4, 2013,
Meielisalp, Lake Thune Switzerland
Contents

1 Overview
2 R Resources
3 Risk-Parity/Budget
4 Optimal Draw Down
5 Probabilistic Utility
6 Optimal Risk/Reward
7 Summary
8 Bibliography
Overview

1. Overview
2. R Resources
3. Risk-Parity/Budget
4. Optimal Draw Down
5. Probabilistic Utility
6. Optimal Risk/Reward
7. Summary
8. Bibliography
Seminal work by Markowitz (1952), i.e., ‘Modern Portfolio Theory’.
Since then, advances in terms of
1. estimators for population parameters.
2. optimization methods.
In general:
return-risk space $\neq$ mean-variance space.
Purpose of this talk: Selective survey of more recent portfolio optimization techniques and how these can be utilized in R.
1 Overview

2 R Resources

3 Risk-Parity/Budget

4 Optimal Draw Down

5 Probabilistic Utility

6 Optimal Risk/Reward

7 Summary

8 Bibliography
R Resources I
Knowing your friends

- **Solver-related R packages:**
  - DEoptim (Mullen et al., 2011), glpkAPI (Gelius-Dietrich, 2012),
  - limSolve (Soetaert et al., 2009), linprog (Henningsen, 2010), lpSolve
    (Berkelaar, 2011), lpSolveAPI (Konis, 2011), quadprog (Turlach and
    Weingessel, 2011), RcppDE (Eddelbuettel, 2012), Rglpk (Theussl and
    Hornik, 2012), rneos (Pfaff, 2011), Rsocp (Chalabi and Würtz, 2010),
    Rsolnp (Ghalanos and Theussl, 2012; Ye, 1987), Rsymphony (Harter
    et al., 2012)

- **Portfolio-related R packages:**
  - fPortfolio (Würtz et al., 2010a), fPortfolioBacktest (Würtz et al.,
    2010b), FRAPO (Pfaff, 2012), parma (Ghalanos, 2013),
    PerformanceAnalytics (Carl et al., 2012), PortfolioAnalytics (Boudt
    et al., 2011b), rportfolios (Novomestky, 2012), tawny (Rowe, 2012)
This should be viewed as a ‘selective’ summary of R packages, there are more! Hence, check CRAN Task Views on ‘Finance’ and ‘Optimization’ and R-Forge for what is available else and for recent additions.

In a nutshell: All kind of portfolio optimization tasks can be accomplished from/within R.
Overview

R Resources

Risk-Parity/Budget

Optimal Draw Down

Probabilistic Utility

Optimal Risk/Reward

Summary

Bibliography
Risk-Parity/Budget

Motivation

- Characteristic: Diversification directly applied to the portfolio risk itself.

- Motivation: Empirical observation that the risk contributions are a good predictor for actual portfolio losses. Hence, portfolio losses can potentially be limited compared to an allocation which witnesses a high risk concentration on one or a few portfolio constituents.

- Risk concepts:
  2. Downside-based, *i.e.*, CVaR/ES (see Boudt et al., 2007, 2008; Peterson and Boudt, 2008; Boudt et al., 2010, 2011a; Ardia et al., 2010), budgeting (BCC) or min-max (MRC).
Risk-Parity/Budget

Problem Delineation

- Starting point general definition of risk contribution:
  \[ C_i M_{\omega \in \Omega} = \omega_i \frac{\partial M_{\omega \in \Omega}}{\partial \omega_i} \]  
  whereby \( M_{\omega \in \Omega} \) signify a linear homogeneous risk measure and \( \omega_i \) is the weight of the i-th asset.

- For volatility-based risk measure:
  \[ \frac{\partial \sigma(\omega)}{\partial \omega_i} = \omega_i \sigma_i^2 + \sum_{i \neq j}^N \omega_j \sigma_{ij} \sigma(\omega) \]  

- For downside-based risk measure:
  \[ C_i \text{CVaR}_{\omega \in \Omega, \alpha} = \omega_i \left[ \mu_i + \frac{(\Sigma \omega)_i}{\sqrt{\omega' \Sigma \omega'}} \frac{\phi(z_\alpha)}{\alpha} \right] \]  
  whereby \( \alpha \) signify the confidence level pertinent to the downside risk.
Example Risk-Parity vs GMV: R Code

```r
## Loading data and computing returns
library(FRAPO)
library(Rsolnp)
data(MultiAsset)
R <- returnseries(MultiAsset, percentage = TRUE, trim = TRUE)

## GMV
wGmvAll <- Weights(PGMV(R))

## ERC for all assets
SigmaAll <- cov(R)
wErcAll <- Weights(PERC2(SigmaAll))

## Two-step, by asset class
SigmaEq <- cov(R[, 1:6])
wErcEq <- Weights(PERC2(SigmaEq))
rEq <- apply(R[, 1:6], 1, function(x) sum(x * wErcEq / 100))
SigmaBd <- cov(R[, 7:9])
wErcBd <- Weights(PERC2(SigmaBd))
rBd <- apply(R[, 7:9], 1, function(x) sum(x * wErcBd / 100))
rAsset <- cbind(rEq, rBd, R[, 10])
SigmaCl <- cov(rAsset)
wErcCl <- Weights(PERC2(SigmaCl))
wErcTwoStage <- c(wErcCl[1] * wErcEq / 100, wErcCl[2] * wErcBd / 100, wErcCl[3])

## comparing the two approaches
W <- cbind(wGmvAll, wErcAll, wErcTwoStage)
Concentration <- apply(W, 2, function(x) sum((x / 100)^2))
```
## Example Risk-Parity vs GMV: Results

<table>
<thead>
<tr>
<th>Assets</th>
<th>GmvAll</th>
<th>ErcAll</th>
<th>ErcTwoStage</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equity</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S&amp;P 500</td>
<td>4.55</td>
<td>3.72</td>
<td>3.06</td>
</tr>
<tr>
<td>Russell 3000</td>
<td>0.00</td>
<td>3.59</td>
<td>2.92</td>
</tr>
<tr>
<td>DAX</td>
<td>4.69</td>
<td>3.47</td>
<td>2.57</td>
</tr>
<tr>
<td>FTSE 100</td>
<td>0.00</td>
<td>4.12</td>
<td>3.45</td>
</tr>
<tr>
<td>Nikkei 225</td>
<td>1.35</td>
<td>3.38</td>
<td>2.68</td>
</tr>
<tr>
<td>MSCI EM</td>
<td>0.00</td>
<td>2.14</td>
<td>1.92</td>
</tr>
<tr>
<td>( \sum \omega^\text{Equity}_i )</td>
<td>10.59</td>
<td>20.43</td>
<td>16.60</td>
</tr>
<tr>
<td><strong>Bond</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>US Treasury</td>
<td>0.00</td>
<td>16.42</td>
<td>18.57</td>
</tr>
<tr>
<td>German REX</td>
<td>88.72</td>
<td>42.44</td>
<td>31.97</td>
</tr>
<tr>
<td>UK Gilts</td>
<td>0.40</td>
<td>15.93</td>
<td>21.37</td>
</tr>
<tr>
<td>( \sum \omega^\text{Bond}_i )</td>
<td>89.12</td>
<td>74.79</td>
<td>71.91</td>
</tr>
<tr>
<td><strong>Commodity</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gold</td>
<td>0.29</td>
<td>4.78</td>
<td>11.49</td>
</tr>
<tr>
<td><strong>Concentration</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \sum \omega^2_i )</td>
<td>0.79</td>
<td>0.24</td>
<td>0.20</td>
</tr>
</tbody>
</table>

**Table:** ERC vs GMV Allocation
library(PortfolioAnalytics)

## Defining constraints and objective for CVaR budget
C1 <- constraint(assets = colnames(R), min = rep(0, N),
                 max = rep(1, N), min_sum = 1, max_sum = 1)
ObjCVaR <- add.objective(constraints = C1, type = "risk", name = "ES",
                         arguments = list(p = 0.95),
                         enabled = TRUE)
ObjCVaRBudget <- add.objective(constraints = ObjCVaR, type = "risk_budget",
                                name = "ES", max_prisk = 0.2, arguments = list(p = 0.95),
                                enabled = TRUE)
SolCVaRBudget <- optimize.portfolio(R = R,
                                     constraints = ObjCVaRBudget, optimize_method = "DEoptim",
                                     itermax = 50, search_size = 20000, trace = TRUE)
WCVaRBudget <- SolCVaRBudget$weights
CVaRBudget <- ES(R, weights = WCVaRBudget, p = 0.95,
                     portfolio_method = "component")

## Minimum CVaR concentration portfolio
ObjCVaRMinCon <- add.objective(constraints = ObjCVaR, type = "risk_budget",
                                name = "ES", min_concentration= TRUE, arguments = list(p = 0.95),
                                enabled = TRUE)
SolCVaRMinCon <- optimize.portfolio(R = R,
                                     constraints = ObjCVaRMinCon, optimize_method = "DEoptim",
                                     itermax = 50, search_size = 20000, trace = TRUE)
WCVaRMinCon <- SolCVaRMinCon$weights
CVaRMinCon <- ES(R, weights = WCVaRMinCon, p = 0.95, portfolio_method = "component")
### Risk-Parity/Budget

#### Example BCC and MRC vs GMV: R Code

<table>
<thead>
<tr>
<th>Assets</th>
<th>Weights</th>
<th></th>
<th>Risk-Contributions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GMV</td>
<td>ERC</td>
<td>BCC</td>
<td>GMV</td>
</tr>
<tr>
<td>S&amp;P 500</td>
<td>4.55</td>
<td>3.72</td>
<td>5.84</td>
<td>2.02</td>
</tr>
<tr>
<td>Russell 3000</td>
<td>0.00</td>
<td>3.59</td>
<td>2.42</td>
<td>1.01</td>
</tr>
<tr>
<td>DAX</td>
<td>4.69</td>
<td>3.47</td>
<td>10.74</td>
<td>1.01</td>
</tr>
<tr>
<td>FTSE 100</td>
<td>0.00</td>
<td>4.12</td>
<td>15.85</td>
<td>4.04</td>
</tr>
<tr>
<td>Nikkei 225</td>
<td>1.35</td>
<td>3.38</td>
<td>2.90</td>
<td>1.01</td>
</tr>
<tr>
<td>MSCI EM</td>
<td>0.00</td>
<td>2.14</td>
<td>5.72</td>
<td>1.01</td>
</tr>
<tr>
<td>US Treasury</td>
<td>0.00</td>
<td>16.42</td>
<td>15.45</td>
<td>18.18</td>
</tr>
<tr>
<td>German REX</td>
<td>88.72</td>
<td>42.44</td>
<td>18.11</td>
<td>66.67</td>
</tr>
<tr>
<td>UK Gilts</td>
<td>0.40</td>
<td>15.93</td>
<td>13.95</td>
<td>1.01</td>
</tr>
<tr>
<td>Gold</td>
<td>0.29</td>
<td>4.78</td>
<td>9.03</td>
<td>4.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GMV</td>
<td>ERC</td>
<td>BCC</td>
<td>MCC</td>
</tr>
<tr>
<td>S&amp;P 500</td>
<td>9.83</td>
<td>16.63</td>
<td>12.73</td>
<td>6.53</td>
</tr>
<tr>
<td>Russell 3000</td>
<td>0.00</td>
<td>16.80</td>
<td>5.57</td>
<td>3.30</td>
</tr>
<tr>
<td>DAX</td>
<td>12.19</td>
<td>14.34</td>
<td>18.98</td>
<td>3.20</td>
</tr>
<tr>
<td>FTSE 100</td>
<td>0.00</td>
<td>11.20</td>
<td>19.94</td>
<td>10.10</td>
</tr>
<tr>
<td>Nikkei 225</td>
<td>3.16</td>
<td>22.36</td>
<td>8.99</td>
<td>4.12</td>
</tr>
<tr>
<td>MSCI EM</td>
<td>0.00</td>
<td>14.22</td>
<td>18.65</td>
<td>5.36</td>
</tr>
<tr>
<td>US Treasury</td>
<td>0.00</td>
<td>5.40</td>
<td>2.31</td>
<td>18.88</td>
</tr>
<tr>
<td>German REX</td>
<td>74.75</td>
<td>−17.60</td>
<td>−3.42</td>
<td>39.61</td>
</tr>
<tr>
<td>UK Gilts</td>
<td>0.50</td>
<td>5.00</td>
<td>0.49</td>
<td>1.18</td>
</tr>
<tr>
<td>Gold</td>
<td>−0.43</td>
<td>11.63</td>
<td>15.78</td>
<td>7.70</td>
</tr>
</tbody>
</table>
Overview

R Resources

Risk-Parity/Budget

Optimal Draw Down

Probabilistic Utility

Optimal Risk/Reward

Summary

Bibliography
Optimal Draw Down

Definition

The draw-down of a portfolio at time $t$ is defined as the difference between the maximum uncompounded portfolio value prior to $t$ and its value at time period $t$. More formally, let $W(\omega, t) = y'_t \omega$ signify the uncompounded portfolio value at time $t$ and $\omega$ are the portfolio weights for the $N$ assets included in it and $y_t$ the cumulated returns, then the draw-down, $D(\omega, t)$, is defined as:

$$D(\omega, t) = \max_{0 \leq \tau \leq t} \{ W(\omega, \tau) \} - W(\omega, t)$$  \hspace{1cm} (4)

The draw down is such a functional risk measure.
Optimal Draw Down
Problem Formulations

- With respect to portfolio optimization, the following problem formulations have been introduced by Chekhlov et al. (2000, 2003, 2005):
  1. Maximum draw down (MaxDD)
  2. Average draw down (AvDD)
  3. Conditional draw down at risk (CDaR)

- The three portfolio optimization approaches can be formulated as a linear program (maximizing average annualized returns and draw downs are included as constraints).

- Implemented in package FRAPO as functions PMaxDD(), PAvDD() and PCDaR(), respectively.
Optimal Draw Down

LP: Maximum Draw Down

The linear program for the MaxDD is given as:

\[
P_{\text{MaxDD}} = \arg \max_{\omega \in \Omega, u \in \mathbb{R}} R(\omega) = \frac{1}{d} y^T \omega
\]

\[
u_1 C \leq u_k - y'_k \omega \leq u_k + y'_k \omega
\]

\[
u_0 = 0
\]

whereby the maximum allowed draw down in nominal terms is defined as a fraction of the available capital/wealth \((\nu_1 C)\) and \(u\) signify a \((T + 1 \times 1)\) vector of slack variables in the program formulation, i.e., the maximum portfolio values up to time period \(k\) with \(1 \leq k \leq T\).
Optimal Draw Down

LP: Average Draw Down

Similarly, the linear program for the AveDD is given as:

\[
P_{\text{AvDD}} = \arg \max_{\omega \in \Omega, u \in \mathbb{R}} R(\omega) = \frac{1}{dC} y'_T \omega \\
\frac{1}{T} \sum_{k=1}^{T} (u_k - y'_k \omega) \leq \nu_2 C \\
u_k \geq y'_k \omega \\
u_k \geq u_{k-1} \\
u_0 = 0
\]
Optimal Draw Down

LP: Conditional Draw Down at Risk

\[
P_{\text{CDaR}} = \arg \max_{\omega \in \Omega, u \in \mathbb{R}, z \in \mathbb{R}, \zeta \in \mathbb{R}} R(\omega) = \frac{1}{dC} y'_T \omega
\]

\[
\zeta + \frac{1}{(1 - \alpha) T} \sum_{k=1}^{T} z_k \leq \nu_3 C
\]

\[
z_k \geq u_k - y'_k \omega - \zeta
\]

\[
z_k \geq 0
\]

\[
u_k \geq y'_k \omega
\]

\[
u_k \geq u_{k-1}
\]

\[
u_0 = 0
\]

whereby \( \zeta \) signify the threshold draw-down value dependent on the prior chosen confidence level \( \alpha \) and the \( (T \times 1) \) vector \( z \) represent the threshold exceedances.
```r
> library(FRAPO)
> library(fPortfolio)
> library(PerformanceAnalytics)
> ## Loading of data set
> data(EuroStoxx50)
> ## Creating timeSeries of prices and returns
> pr <- timeSeries(EuroStoxx50, charvec = rownames(EuroStoxx50))
> NAssets <- ncol(pr)
> RDP <- na.omit((pr / lag(pr, k = 1) - 1) * 100)
> ## Backtest of GMV vs. CDaR
> ## Start and end dates
> to <- time(RDP)[208:nrow(RDP)]
> from <- rep(start(RDP), length(to))
> ## Portfolio specifications
> ## CDaR portfolio
> DDbound <- 0.10
> DDalpha <- 0.95
> ## GMV portfolio
> mvspec <- portfolioSpec()
> BoxC <- c("minsumW[1:NAssets] = 0.0", "maxsumW[1:NAssets] = 1.0")
> ## Initialising weight matrices
> wMV <- wCD <- matrix(NA, ncol = ncol(RDP), nrow = length(to))
> ## Conducting backtest
> for(i in 1:length(to)){
+   series <- window(RDP, start = from[i], end = to[i])
+   prices <- window(pr, start = from[i], end = to[i])
+   mv <- minvariancePortfolio(data = series, spec = mvspec, constraints = BoxC)
```
Optimal Draw Down II

Example Stock Portfolio: GMV vs. CDaR

```r
+  cd <- PCDaR(prices, alpha = DDalpha, bound = DDbound, softBudget = TRUE)
+  wMV[i, ] <- c(getWeights(mv))
+  wCD[i, ] <- Weights(cd)
+ }

## Lagging optimal weights and sub-sample of returns
> wMV <- rbind(rep(NA, ncol(RDP)), wMV[-nrow(wMV), ])
> wMV <- timeSeries(wMV, charvec = to)
> colnames(wMV) <- colnames(RDP)

## Portfolio equities of strategies
> wCD <- rbind(rep(NA, ncol(RDP)), wCD[-nrow(wCD), ])
> wCD <- timeSeries(wCD, charvec = to)
> colnames(wCD) <- colnames(RDP)

> ## Portfolio returns
> MVRetFac <- 1 + rowSums(wMV1 * RDPback) / 100
> MVRetFac[1] <- 100
> MVPort <- timeSeries(cumprod(MVRetFac), charvec = names(MVRetFac))
> CDRetFac <- 1 + rowSums(wCDL1 * RDPback) / 100
> CDRetFac[1] <- 100
> CDPort <- timeSeries(cumprod(CDRetFac), charvec = names(CDRetFac))

## Draw down table
> table.Drawdowns(MVRet)
> table.Drawdowns(CDRet)
```
## Optimal Draw Down

### Example Stock Portfolio: GMV vs. CDaR

<table>
<thead>
<tr>
<th>Portfolio</th>
<th>From</th>
<th>Trough</th>
<th>To</th>
<th>Depth</th>
<th>→</th>
<th>↓</th>
<th>↑</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GMV</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2007-12-10</td>
<td>2008-03-17</td>
<td>NA</td>
<td>20.11</td>
<td>17</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2007-06-04</td>
<td>2007-08-13</td>
<td>2007-10-08</td>
<td>9.75</td>
<td>19</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>2007-03-12</td>
<td>2007-03-12</td>
<td>2007-03-19</td>
<td>2.30</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2007-04-23</td>
<td>2007-04-23</td>
<td>2007-04-30</td>
<td>0.76</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

| **CDaR**  |            |            |             |       |   |   |   |
| 1         | 2007-11-12 | 2008-01-21 | NA          | 11.53 | 21| 11|   |
| 2         | 2007-06-04 | 2007-09-03 | 2007-10-08  | 5.58  | 19| 14| 5 |
| 3         | 2007-05-07 | 2007-05-07 | 2007-05-14  | 0.51  | 2 | 1 | 1 |
| 4         | 2007-03-12 | 2007-03-12 | 2007-03-19  | 0.49  | 2 | 1 | 1 |
| 5         | 2007-10-22 | 2007-10-29 | 2007-11-05  | 0.30  | 3 | 2 | 1 |

**Table:** Overview of Draw Downs (positive, percentages)
Probabilistic Utility

Motivation

- Portfolio selection problems derived from utility functions.
- E.g. mean-variance optimisation:
  \[ U = \lambda \omega' \mu - (1 - \lambda) \omega' \Sigma \omega. \]
- Allocation sensitive to parameters \( \mu, \Sigma, \lambda \).
- Problem-solving approaches: robust/bayesian estimators and/or robust optimization.
- Nota bene: \( \mu \) and \( \Sigma \) are random variables; as such the allocation vector \( \omega \) is a random variable itself.
- Now: probabilistic interpretation of utility functions.
Probabilistic Utility

Concept 1

- Approach introduced by Rossi et al. (2002) and Marschinski et al. (2007).
- Utility function is interpreted as the logarithm of the probability density for a portfolio.
- Optimal allocation is defined as the expected value of the portfolio’s weights with respect to that probability, i.e., the weights are viewed as parameters of this distribution.
Given: \( u = u(\omega, U, \theta) \), whereby \( \omega \) is weight vector, \( U \) the assumed utility function and \( \theta \) a catch-all parameter vector (e.g. expected returns, dispersion, risk sensitivity).

Expected utility is proportional to the logarithm of a probability measure:
\[
\omega \sim P(\omega|U, \theta) = Z^{-1}(\nu, U, \theta) \exp (\nu u(\omega, U, \theta)).
\]

Normalizing constant: \( Z(\nu, U, \theta) = \int_{\mathcal{D}(\omega)}[d\omega] \exp (\nu u(\omega, U, \theta)) \).

Convergence to maximum utility (\( \nu \to \infty \)) or equal-weight solution (\( \nu \to 0 \)) is controlled by: \( \nu = pN^\gamma \).

Portfolio solution is then defined as:
\[
\bar{\omega}(U, \theta) = Z^{-1}(\nu, U, \theta) \int_{\mathcal{D}(\omega)}[d\omega] \omega \exp (\nu u(\omega, U, \theta)).
\]
## Utility function

```r
U1 <- function(x, mu, risk, lambda = 0.5){
  lambda * x * mu - (1 - lambda) * risk * x^2
}
```

## Sequence of possible weights

```r
x <- seq(0, 1, length.out = 1000)
```

## Utility

```r
u1 <- U1(x, mu = 5, risk = 16, lambda = 0.5)
```

## Optimal Allocation (in percentage)

```r
MUopt <- round(x[which.max(u1)] * 100, 2)
```

## Now introducing concept of probabilistic utility

```r
U1DU <- function(x, mu, risk, lambda = 0.5, nu = 1){
  exp(nu * U1(x = x, mu = mu, risk = risk, lambda = lambda))
}
```

```r
u1u <- U1DU(x, mu = 5, risk = 16, lambda = 0.5, nu = 1)
```

## Density

```r
U1DS <- function(x, mu, risk, lambda = 0.5, nu = 1){
  Dconst <- integrate(U1DU, lower = 0, upper = 1, mu = mu,
    risk = risk, lambda = lambda, nu = nu)$value
  1 / Dconst * U1DU(x = x, mu = mu, risk = risk, lambda = lambda, nu = nu)
}
```

## Compute expected value as optimal weight for risky asset

```r
PUopt <- round(mean(x * U1DS(x = x, mu = 5, risk = 16, lambda = 0.5, nu = 1)) * 100, 2)
```

## Associated utility

```r
U1MU <- U1(MUopt / 100, mu = 2, risk = 9, lambda = 0.5)
U1PU <- U1(PUopt / 100, mu = 2, risk = 9, lambda = 0.5)
```
Probabilistic Utility

Example: quadratic utility, one risky asset, II
Probabilistic Utility

Example: quadratic utility, one risky asset, III
Probabilistic Utility
Markov Chain Monte Carlo

- Class of algorithms for sampling from a probability distribution; shape of density suffices.
- Purpose of MCMC is the numeric evaluation of multi-dimensional integrals, by (i) searching and (ii) evaluating the state space.
- The state space is searched by means of a Markov chain-type progression of the parameters.
- Evaluating proposed move (accepting/rejecting) ordinarily by Metropolis-Hastings algorithm.
- R resources: numerous R packages are available; see CRAN and task view ‘Bayesian’ for an annotated listing.
- Book resources: Gilks et al. (1995) and Brooks et al. (2011).
Probabilistic Utility

Hybrid Monte Carlo

- Introduced by Duane et al. (1987) (see Neal (2011) for a more textbook-like exposition).
- Inclusion of an auxiliary momentum vector and taking the gradient of the target distribution into account.
- Purpose/aim:
  1. Moving through state space in larger steps.
  2. Autocorrelation in Markov Chains less pronounced compared to other approaches (thinning in principal not necessary).
  3. High acceptance rate, ideally all moves are accepted.
  4. Faster convergence to equilibrium distribution.
Probabilistic Utility

Hybrid Monte Carlo II

- Amending density by conjugate variables $p$:

$$G(q, p) \sim \exp \left( U(q) - \frac{p'p}{2} \right)$$  \hspace{1cm} (8)

- Algorithm: Starting from a pair $(q_n, p_n)$
  1. Sample $\eta$ from standard normal.
  2. For a time interval $T$, integrate Hamiltonian equations:

$$\frac{dp_i}{dt} = -\frac{\delta U}{\delta p_i} \hspace{1cm} (9a)$$
$$\frac{dq_i}{dt} = p_i \hspace{1cm} (9b)$$

  together with the boundary constraints $p(0) = \eta$ and $q(0) = q_n$.

  3. Accept $q_{n+1} = q(T)$ with probability:

$$\beta = \min(1, \exp (G(q(T), p(T)) - G(q_n, \eta))) \hspace{1cm} (10)$$

  else set $q_{n+1} = q_n$.
Probabilistic Utility

Hybrid Monte Carlo III

See http://www.cs.utoronto.ca/~radford/GRIMS.html (adopted version)

hybridMC <- function(logDens, cState, eps, L, ...){
  q <- cState
  p <- rnorm(length(q), 0, 1) ## independent standard normal variates
  cMom <- p
  ## Make a half step for momentum at the beginning
  p <- p + eps * grad(func = logDens, x = q, ...) / 2
  ## Alternate full steps for position and momentum
  for (i in 1:L){
    ## Make a full step for the position
    q <- q + eps * p
    ## Check lower bound
    lbidx <- which(q < 0)
    if(length(lbidx) > 0){
      q[lbidx] <- -q[lbidx]
      p[lbidx] <- -p[lbidx]
    }
    ## Check budget constraint
    qsum <- sum(q)
    q <- q / qsum
    ## Make a full step for the momentum, except at end of trajectory
    if (i!=L) p <- p + eps * grad(func = logDens, x = q, ...)
  }
  ## Make a half step for momentum at the end.
  p <- p + eps * grad(func = logDens, x = q, ...) / 2
  ## Negate momentum at end of trajectory to make the proposal symmetric
  p <- -p
## Evaluate potential and kinetic energies at start and end of trajectory

clogDens <- logDens(cState, ...)
cK <- sum(cMom^2) / 2
Hinit <- pexp(clogDens - cK)
plogDens <- logDens(q, ...)
pK <- sum(p^2) / 2
Hprop <- pexp(plogDens - pK)

## Accept or reject the state at end of trajectory, returning either
## the position at the end of the trajectory or the initial position

delta <- Hprop - Hinit
apr <- min(1, exp(delta))
ifelse(runif(1) < apr, return(q), return(cState))

## Quadratic Utility Function

U <- function(x, mu, Sigma, lambda = 0.5){
  c(lambda * t(x) %*% mu) - c((1 - lambda) * t(x) %*% Sigma %*% x)
}

## Log-density of quadratic utility
LUdens <- function(x, mu, Sigma, lambda = 0.5, nu){
  nu * U(x = x, mu = mu, Sigma = Sigma, lambda = lambda)
}

## Expected utility of Quadratic Utility Function
PUopt <- function(logDens, MCSteps, BurnIn, eps, L, mu, Sigma, lambda = 0.5, nu){
  J <- length(mu)
  MCMC <- matrix(NA, ncol = J, nrow = MCSteps)
  MCMC[1, ] <- rep(1/J, J)
  for(i in 2:MCSteps){
    # code continues here...
Probabilistic Utility

Probabilistic Utility III
Hybrid Monte Carlo III

```r
MCMC[i, ] <- hybridMC(logDens = logDens, cState = MCMC[i - 1, ],
                      eps = eps, L = L, mu = mu, Sigma = Sigma,
                      lambda = lambda, nu = nu)
}
MCMC <- MCMC[-c(1:BurnIn), ]
MCMC
}

## Maximization of Quadratic Utility Function
MUopt <- function(mu, Sigma, lambda){
  V <- (1 - lambda) * 2 * Sigma
  N <- ncol(Sigma)
  a1 <- rep(1, N)
  b1 <- 1
  a2 <- diag(N)
  b2 <- rep(0, N)
  Amat <- cbind(a1, a2)
  Bvec <- c(b1, b2)
  meq <- c(1, rep(0, N))
  opt <- solve.QP(Dmat = V, dvec = lambda * mu, Amat = Amat, bvec = Bvec, meq = meq)
  opt$solution
}
```
Michaud-type simulation (see Michaud, 1989, 1998) as in Marschinski et al. (2007):

1. Treat estimates of location and dispersion as true population parameters for a given sample.
2. Obtain optimal ‘true’ MU allocations and hence utility.
3. Draw $K$ random samples of length $L$ from these ‘population’ parameters and obtain MU and PU solutions.
4. Compare distances of these $K$ solutions with ‘true’ utility.

Settings: Sample sizes ($L$) of 24, 30, 36, 48, 54, 60, 72, 84, 96, 108 and 120 observations; length of MC 250 (150 burn-in-periods) and $K$ equals 100.

## Load packages
library(FRAPO)
library(MASS)
library(numDeriv)
library(parallel)
library(compiler)
enableJIT(3)
## Loading data and computing returns
data(MultiAsset)
Assets <- timeSeries(MultiAsset, charvec = rownames(MultiAsset))
R <- returns(Assets, method = "discrete", percentage = TRUE)
J <- ncol(R)
N <- nrow(R)
## Population moments, max util weights and utility
MuPop <- apply(R, 2, mean)
SigmaPop <- cov(R)
WeightsPop <- MUopt(m = MuPop, S = SigmaPop, lambda = 0.9)
UtilPop <- U(WeightsPop, mu = MuPop, Sigma = SigmaPop, lambda = 0.9)
## Parameters and initialising of simulation
Draws <- 100
Idx <- 1:Draws
Samples <- c(24, 30, 36, 48, 54, 60, 72, 84, 96, 108, 120)
LS <- length(Samples)
PU <- matrix(NA, ncol = LS, nrow = Draws)
MU <- matrix(NA, ncol = LS, nrow = Draws)
colnames(PU) <- colnames(MU) <- paste("S", Samples, sep = ")
### Probabilistic Utility II

#### Comparative Simulation: R Code

```r
PUW <- array(NA, dim = c(Draws, J, LS))
MUW <- array(NA, dim = c(Draws, J, LS))

## Parallel processing
cl <- makeCluster(3)
clusterExport(cl = cl, c("MUopt", "PUopt", "solve.QP", "U", "hybridMC", "grad", "LUdens"))

## Utility simulation: function for computing and evaluating MU and PU
Util <- function(x, MCSteps, BurnIn, eps, L, lambda, nu, MuPop, SigmaPop){
  J <- ncol(x)
  mu <- apply(x, 2, mean)
  sigma <- cov(x)
  ## Max Utility for sample weights, with population moments
  MUW <- MUopt(mu, sigma, lambda)
  MU <- U(MUW, MuPop, SigmaPop, lambda)
  ## Prob Utility for sample weights, with population moments
  MCMC <- PUopt(LUdens, MCSteps, BurnIn, eps, L, mu, sigma, lambda, nu)
  PUW <- colMeans(MCMC)
  PU <- U(PUW, MuPop, SigmaPop, lambda)
  list(U = c(MU, PU), PUW = PUW, MUW = MUW)
}
```
for(i in 1:LS){
  cat(paste("Computing for Sample Size", Samples[i], ", \n"))
  SampleL <- Samples[i]
  ListData <- lapply(Idx, function(x) mvrnorm(n = SampleL, mu = MuPop, Sigma = SigmaPop))
  MuPu <- parLapplyLB(cl = cl, ListData, Util, MCSteps = 250, BurnIn = 150,
    eps = 1 / SampleL, L = SampleL,
    lambda = 0.9, nu = SampleL, MuPop = MuPop, SigmaPop = SigmaPop)
  MU[, i] <- unlist(lapply(MuPu, function(x) x$U[1]))
  PU[, i] <- unlist(lapply(MuPu, function(x) x$U[2]))
  PUW[, , i] <- matrix(unlist(lapply(MuPu, function(x) x$PUW)), ncol = ncol(R), nrow = Draws, byrow = TRUE)
  MUW[, , i] <- matrix(unlist(lapply(MuPu, function(x) x$MUW)), ncol = ncol(R), nrow = Draws, byrow = TRUE)
}
stopCluster(cl = cl)
## Computing distances
## Computing distances
MUD <- (UtilPop - MU) / UtilPop * 100
PUD <- (UtilPop - PU) / UtilPop * 100
Probabilistic Utility

Comparative Simulation: R Code, Distances from true utility
1 Overview
2 R Resources
3 Risk-Parity/Budget
4 Optimal Draw Down
5 Probabilistic Utility
6 **Optimal Risk/Reward**
7 Summary
8 Bibliography
Optimal Risk/Reward

Definition

- Fractional (non-)linear programming problem:

\[
P_{\text{Ratio}} = \arg \min_{\omega \in \Omega} \frac{f_{\text{Risk}}(R, \omega, \theta)}{f_{\text{Reward}}(R, \omega)}
\]

\[
\omega' i = 1
\]

\[
\omega \geq 0
\]

\[
l \leq A \omega \leq u
\]

- Key developments by Charnes and Cooper (1969) (linear case) and Dinkelbach (1967); Schaible (1967a,b); Stoyanov et al. (2007) (non-linear case).

- Risk measures: Variance, MAD, minimizing maximum loss, lower partial moment, CVaR, CDaR.
Using the semi-standard deviation as risk-measure has been mentioned in Markowitz (1952).

The lower partial moment is defined as:

$$\text{LPM}_{n,\tau} = \int_{-\infty}^{\tau} (\tau - x)^n f(x) dx,$$  \hspace{1cm} (12)

whereby $x$ is the random variable, $f(x)$ the associated density function, $\tau$ is the target for which the deviations are measured and $n$ signify the weighting of the deviations from the threshold.

The semi-variance results as a special for $\tau = E(x)$ and $n = 2$. 
Optimal Risk/Reward I

Optimal portfolio with LPM: R Code

```r
> library(parma)
> rlpm <- parmaspec(scenario = R, forecast = colMeans(R),
+ risk = "LPM", target = mean(colMeans(R)), targetType = "equality",
+ riskType = "optimal", options = list(threshold = 999, moment = 2),
+ LB = rep(0, 10), UB = rep(1, 10), budget = 1)
> parmasolve(rlpm, type = "NLP")

+---------------------------------+
| PARMA Portfolio                  |
+---------------------------------+
| No.Assets : 10                   |
| Problem : NLP                    |
| Risk Measure : LPM               |
| Objective : optimal              |
| Risk : 0.6766982                 |
| Reward : 0.5376806               |

Optimal_Weights
GREXP 0.8176
GLD 0.1019
GDAXI 0.0805

> ## Charming outcome: Allocate to German Bonds & Equity and Gold :-)
1 Overview

2 R Resources

3 Risk-Parity/Budget

4 Optimal Draw Down

5 Probabilistic Utility

6 Optimal Risk/Reward

7 Summary

8 Bibliography
More than sixty years after seminal work of Markowitz, progress has centred on how the risk-return space is modeled.

Advances were driven by financial market crisis.

Basically, all of these newly proposed portfolio optimization approaches can addressed within/from R.

In a kaleidoscopic fashion, some of these advances have been introduced in this talk, but . . .
...more examples in...


Bibliography II


