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

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

- 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)

R Resources II

Knowing your friends

- 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.

1 Overview

2 R Resources

3 Risk-Parity/Budget

4 Optimal Draw Down

5 Probabilistic Utility

6 Optimal Risk/Reward

7 Summary

8 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:
  1. Volatility-based, *i.e.* standard deviation (see Qian, 2005, 2006, 2011; Maillard et al., 2009, 2010)
  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:

$$C_i M_{\omega \in \Omega} = \omega_i \frac{\partial M_{\omega \in \Omega}}{\partial \omega_i} \quad (1)$$

whereby $M_{\omega \in \Omega}$ signify a linear homogeneous risk measure and ω_i is the weight of the i-th asset.

- For volatility-based risk measure:

$$\frac{\partial \sigma(\omega)}{\partial \omega_i} = \frac{\omega_i \sigma_i^2 + \sum_{i \neq j}^{N} \omega_j \sigma_{ij}}{\sigma(\omega)} \quad (2)$$

- For downside-based risk measure:

$$C_i \mathrm{CVaR}_{\omega \in \Omega, \alpha} = \omega_i \left[\mu_i + \frac{(\Sigma \omega)_i}{\sqrt{\omega' \Sigma \omega'}} \frac{\phi(z_\alpha)}{\alpha} \right] \quad (3)$$

whereby α signify the confidence level pertinent to the downside risk.

Risk-Parity/Budget

Example Risk-Parity vs GMV: R Code

```
> library(FRAPO)
> library(Rsolnp)
> ## Loading data and computing returns
> 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 measure
> Concentration <- apply(W, 2, function(x) sum((x / 100)^2))
```

Risk-Parity/Budget

Example Risk-Parity vs GMV: Results

Assets	GmvAll	ErcAll	ErcTwoStage
Equity			
S&P 500	4.55	3.72	3.06
Russell 3000	0.00	3.59	2.92
DAX	4.69	3.47	2.57
FTSE 100	0.00	4.12	3.45
Nikkei 225	1.35	3.38	2.68
MSCI EM	0.00	2.14	1.92
$\sum \omega_i^{\text{Equity}}$	10.59	20.43	16.60
Bond			
US Treasury	0.00	16.42	18.57
German REX	88.72	42.44	31.97
UK Gilts	0.40	15.93	21.37
$\sum \omega_i^{\text{Bond}}$	89.12	74.79	71.91
Commodity			
Gold	0.29	4.78	11.49
Concentration			
$\sum \omega_i^2$	0.79	0.24	0.20

Table: ERC vs GMV Allocation

Risk-Parity/Budget

Example BCC and MRC vs GMV: R Code

```r
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

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

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 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 $W(\omega, t) = \mathbf{y}_t' \omega$ signify the uncompounded portfolio value at time t and ω are the portfolio weights for the N assets included in it and $\mathbf{y}_t$ the cumulated returns, then the draw-down, $\mathbb{D}(\omega, t)$, is defined as:

$$\mathbb{D}(\omega, t) = \max_{0 \leq \tau \leq t} \{W(\omega, \tau)\} - W(\omega, t) \tag{4}$$

The draw down is as 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()`, `PAveDD()` and `PCDaR()`, respectively.

Optimal Draw Down

LP: Maximum Draw Down

The linear program for the MaxDD is given as:

$$
\begin{aligned}
\mathrm{P}_{\mathrm{MaxDD}} = & \underset{\omega \in \Omega, \mathbf{u} \in \mathbb{R}}{\arg\max} \; R(\omega) = \frac{1}{dC} \mathbf{y}'_T \omega \\
& u_k - \mathbf{y}'_k \omega \leq \nu_1 C \\
& u_k \geq \mathbf{y}'_k \omega \\
& u_k \geq u_{k-1} \\
& u_0 = 0
\end{aligned} \tag{5}
$$

whereby the maximum allowed draw down in nominal terms is defined as a fraction of the available capital/wealth ($\nu_1 C$) and $\mathbf{u}$ 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:

$$
\begin{aligned}
\mathrm{P}_{\mathrm{AvDD}} = \underset{\omega \in \Omega, \mathbf{u} \in \mathbb{R}}{\arg\max}\ R(\omega) &= \frac{1}{dC}\mathbf{y}'_T\omega \\
\frac{1}{T}\sum_{k=1}^{T}\left(u_k - \mathbf{y}'_k\omega\right) &\leq \nu_2 C \\
u_k &\geq \mathbf{y}'_k\omega \\
u_k &\geq u_{k-1} \\
u_0 &= 0
\end{aligned}
\tag{6}
$$

Optimal Draw Down

LP: Conditional Draw Down at Risk

$$
\begin{aligned}
\mathrm{P}_{\mathrm{CDaR}} = \underset{\omega \in \Omega, \mathbf{u} \in \mathbb{R}, \mathbf{z} \in \mathbb{R}, \zeta \in \mathbb{R}}{\arg\max} \quad & R(\omega) = \frac{1}{dC} \mathbf{y}'_T \omega \\
& \zeta + \frac{1}{(1-\alpha)T} \sum_{k=1}^{T} z_k \leq \nu_3 C \\
& z_k \geq u_k - \mathbf{y}'_k \omega - \zeta \\
& z_k \geq 0 \\
& u_k \geq \mathbf{y}'_k \omega \\
& u_k \geq u_{k-1} \\
& u_0 = 0
\end{aligned}
\tag{7}
$$

whereby ζ signify the threshold draw-down value dependent on the prior choosen confidence level α and the $(T \times 1)$ vector $\mathbf{z}$ represent the threshold exceedances.

Optimal Draw Down I

Example Stock Portfolio: GMV vs. CDaR

```
> 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

```
+   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), ])
> wMVL1 <- timeSeries(wMV, charvec = to)
> colnames(wMVL1) <- colnames(RDP)
> wCD <- rbind(rep(NA, ncol(RDP)), wCD[-nrow(wCD), ])
> wCDL1 <- timeSeries(wCD, charvec = to)
> colnames(wCDL1) <- colnames(RDP)
> RDPback <- RDP[to,]
> colnames(RDPback) <- colnames(RDP)
> ## Portfolio equities of strategies
> MVRetFac <- 1 + rowSums(wMVL1 * 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))
> ## Portfolio returns
> MVRet <- returns(MVPort, method = "discrete", percentage = FALSE, trim = TRUE)
> CDRet <- returns(CDPort, method = "discrete", percentage = FALSE, trim = TRUE)
> ## Draw down table
> table.Drawdowns(MVRet)
> table.Drawdowns(CDRet)
```

Optimal Draw Down

Example Stock Portfolio: GMV vs. CDaR

Portfolio	From	Trough	To	Depth	$\rightarrow$	$\searrow$	$\nearrow$
GMV							
1	2007-12-10	2008-03-17	NA	20.11	17	15	
2	2007-06-04	2007-08-13	2007-10-08	9.75	19	11	8
3	2007-10-15	2007-11-05	2007-11-26	3.34	7	4	3
4	2007-03-12	2007-03-12	2007-03-19	2.30	2	1	1
5	2007-04-23	2007-04-23	2007-04-30	0.76	2	1	1
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)

1 Overview

2 R Resources

3 Risk-Parity/Budget

4 Optimal Draw Down

5 Probabilistic Utility

6 Optimal Risk/Reward

7 Summary

8 Bibliography

Probabilistic Utility

Motivation

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

Probabilistic Utility

Concept I

- 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.

Probabilistic Utility

Concept II

- Given: $u = u(\boldsymbol{\omega}, U, \theta)$, whereby $\boldsymbol{\omega}$ is weight vector, U the assumed utility function and θ a catch-all parameter vector (*e.g.* expected returns, dispersion, risk sensitivity).
- Expected utility is proportional to the logarithm of a probability measure:
  $\boldsymbol{\omega} \sim P(\boldsymbol{\omega}|U, \theta) = Z^{-1}(\nu, U, \theta) \exp(\nu u(\boldsymbol{\omega}, U, \theta))$.
- Normalizing constant: $Z(\nu, U, \theta) = \int_{\mathfrak{D}(\boldsymbol{\omega})} [\mathrm{d}\boldsymbol{\omega}] \exp(\nu u(\boldsymbol{\omega}, U, \theta))$.
- 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:
  $\bar{\boldsymbol{\omega}}(U, \theta) = Z^{-1}(\nu, U, \theta) \int_{\mathfrak{D}(\boldsymbol{\omega})} [\mathrm{d}\boldsymbol{\omega}] \boldsymbol{\omega} \exp(\nu u(\boldsymbol{\omega}, U, \theta))$

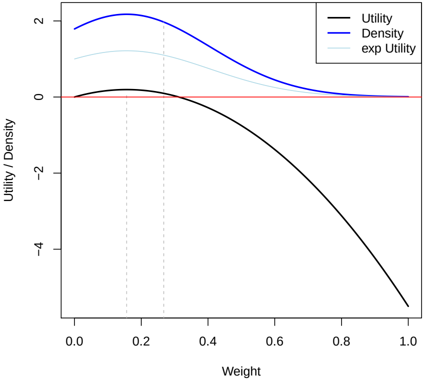
Probabilistic Utility

Example: quadratic utility, one risky asset, I

```
## Utility function
U1 <- function(x, mu, risk, lambda = 0.5){
  lambda * x * mu - (1 - lambda) * risk * x^2
}
## Sequence of possible weights
x <- seq(0, 1, length.out = 1000)
## Utility
u1 <- U1(x, mu = 5, risk = 16, lambda = 0.5)
## Optimal Allocation (in percentage)
MUopt <- round(x[which.max(u1)] * 100, 2)
## Now introducing concept of probabilistic utility
U1DU <- function(x, mu, risk, lambda = 0.5, nu = 1){
  exp(nu * U1(x = x, mu = mu, risk = risk, lambda = lambda))
}
u1u <- U1DU(x, mu = 5, risk = 16, lambda = 0.5, nu = 1)
## Density
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
PUopt <- round(mean(x * U1DS(x = x, mu = 5, risk = 16, lambda = 0.5, nu = 1)) * 100, 2)
## Associated utility
U1MU <- U1(MUopt / 100, mu = 2, risk = 9, lambda = 0.5)
U1PU <- U1(PUopt / 100, mu = 2, risk = 9, lambda = 0.5)
```

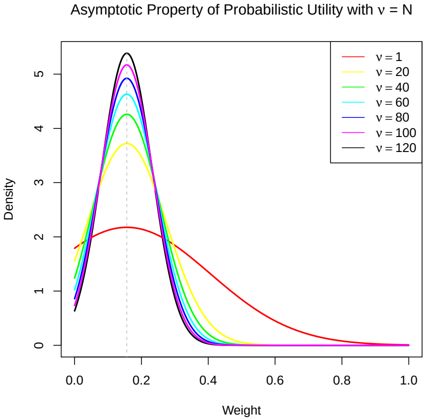
Probabilistic Utility

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 I

- Introduced by Duane et al. (1987) (see Neal (2011) for a more textbook-like exposition).
- Inclusion of an auxilliary 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 $\mathbf{p}$:

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

- Algorithm: Starting from a pair $(\mathbf{q}_n, \mathbf{p}_n)$
  1. Sample $\boldsymbol{\eta}$ from standard normal.
  2. For a time interval T, integrate Hamiltonion equations:

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\delta U}{\delta p_i} \tag{9a}$$

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = p_i \tag{9b}$$

     together with the boundary constraints $\mathbf{p}(0) = \boldsymbol{\eta}$ and $\mathbf{q}(0) = \mathbf{q}_n$.
  3. Accept $\mathbf{q}_{n+1} = \mathbf{q}(T)$ with probability:

$$\beta = min(1, \exp\left(G(\mathbf{q}(T), \mathbf{p}(T)) - G(\mathbf{q}_n, \boldsymbol{\eta})\right)), \tag{10}$$

     else set $\mathbf{q}_{n+1} = \mathbf{q}_n$.

Probabilistic Utility I

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
```

Probabilistic Utility II

Hybrid Monte Carlo III

```
  ## 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)
  delta <- Hprop - Hinit
  ## Accept or reject the state at end of trajectory, returning either
  ## the position at the end of the trajectory or the initial position
  apr <- min(1, exp(delta))
  ifelse(runif(1) < apr, return(q), return(cState))
}
## Quadratic Utility Funtion
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){
```

Probabilistic Utility III

Hybrid Monte Carlo III

```
    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
}
```

Probabilistic Utility

Comparative Simulation: Design

- 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.
- Applied to end-of-month multi-asset data set contained in R package `FRAPO` (see Pfaff, 2012), sample period 2004:11 – 2011:11.

Probabilistic Utility I

Comparative Simulation: R Code

```
## 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

```
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)
}
```

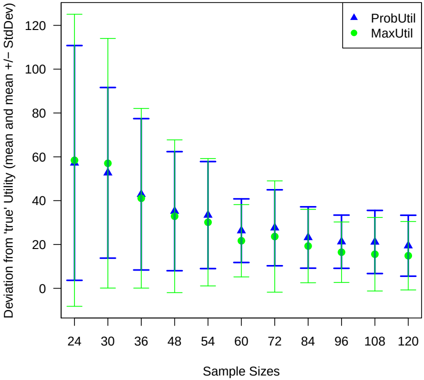
Probabilistic Utility III

Comparative Simulation: R Code

```
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
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:

$$
\begin{aligned}
\mathrm{P}_{\text{Ratio}} = & \arg\min_{\omega \in \Omega} \frac{f_{\text{Risk}}(R, \omega, \theta)}{f_{\text{Reward}}(R, \omega)} \\
& \omega' \mathbf{i} = 1 \\
& \omega \geq 0 \\
& \mathbf{l} \leq A\omega \leq \mathbf{u}
\end{aligned}
\tag{11}
$$

- 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.

Optimal Risk/Reward

Optimal portfolio with LPM: Closing the loop

- Using the semi-standard deviation as risk-measure has been mentioned in Markowitz (1952).
- The lower partial moment is defined as:

$$\mathrm{LPM}_{n,\tau} = \int_{-\infty}^{\tau} (\tau - x)^n f(x) \mathrm{d}x \; , \tag{12}$$

  whereby x is the random variable, $f(x)$ the associated density function, τ 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 $\tau = \mathrm{E}(x)$ and $n = 2$.

Optimal Risk/Reward I

Optimal portfolio with LPM: R Code

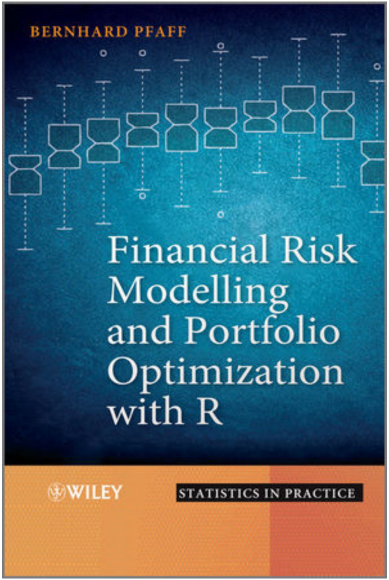
```
> library(parma)
> rlpm <- parmaspec(scenario = R, forecast = colMeans(R),
+  risk = "LPM", target = mean(colMeans(R)), targetType = "equality",
+  riskType = "optimal", options = list(threshold = 999, moment = 2),
+  LB = rep(0, 10), UB = rep(1, 10), budget = 1)
> parmasolve(rlpm, type = "NLP")

+---------------------------------+
|        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

Summary

- 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 . . .



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

Bibliography I

Ardia, D., K. Boudt, P. Carl, K. Mullen, and B. Peterson (2010). Differential Evolution (DEoptim) for non-convex portfolio optimization.

Berkelaar, M. (2011). *lpSolve: Interface to Lp_solve v. 5.5 to solve linear/integer programs*. R package version 5.6.6.

Boudt, K., P. Carl, and B. Peterson (2010, April). Portfolio optimization with cvar budgets. Presentation at r/finance conference, Katholieke Universteit Leuven and Lessius, Chicago, IL.

Boudt, K., P. Carl, and B. Peterson (2011a, September). Asset allocation with conditional value-at-risk budgets. Technical report, http://ssrn.com/abstract=1885293.

Boudt, K., P. Carl, and B. Peterson (2011b). *PortfolioAnalytics: Portfolio Analysis, including Numeric Methods for Optimization of Portfolios*. R package version 0.6.1/r1849.

Boudt, K., B. Peterson, and C. Croux (2007, September). Estimation and decomposition of downside risk for portfolios with non-normal returns. Working Paper KBI 0730, Katholieke Universteit Leuven, Faculty of Economics and Applied Economics, Department of Decision Sciences and Information Management (KBI), Leuven.

Boudt, K., B. Peterson, and C. Croux (2008). Estimation and decomposition of downside risk for portfolios with non-normal returns. *The Journal of Risk 11*(2), 79–103.

Brooks, S., A. Gelman, G. Jones, and X.-L. Meng (Eds.) (2011). *Handbook of Markow Chain Monte Carlo*. Boca Raton, FL: Chapman & Hall / CRC.

Carl, P., B. Peterson, K. Boudt, and E. Zivot (2012). *PerformanceAnalytics: Econometric tools for performance and risk analysis.* R package version 1.0.4.4.

Chalabi, Y. and D. Würtz (2010). *Rsocp: An R extenstion library to use SOCP from R.* R package version 271.1/r4910.

Charnes, A. and W. Cooper (1969). Programming with linear fractional functionals. *Naval Research logistics quarterly 9*(3–4), 181–186.

Chekhlov, A., S. Uryasev, and M. Zabarankin (2000). Portfolio optimization with drawdown constraints. Research report 2000-5, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL.

Bibliography II

Chekhlov, A., S. Uryasev, and M. Zabarankin (2003, January). Portfolio optimization with drawdown constraints. Working paper, University of Florida, Gainesville, FL.

Chekhlov, A., S. Uryasev, and M. Zabarankin (2005). Drawdown measure in portfolio optimization. *International Journal of Theoretical and Applied Finance 8*(1), 13–58.

Dinkelbach, W. (1967). On nonlinear fractional programming. *Management Science 13*(7), 492–498.

Duane, S., A. Kennedy, B. Pendleton, and D. Roweth (1987). Hybrid monte carlo. *Physical Letters B195*, 216–222.

Eddelbuettel, D. (2012). *RcppDE: Global optimization by differential evolution in C++*. R package version 0.1.1.

Gelius-Dietrich, G. (2012). *glpkAPI: R Interface to C API of GLPK*. R package version 1.2.1.

Ghalanos, A. (2013). *parma: portfolio allocation and risk management applications.* R package version 1.03.

Ghalanos, A. and S. Theussl (2012). *Rsolnp: General Non-linear Optimization Using Augmented Lagrange Multiplier Method*. R package version 1.14.

Gilks, W., S. Richardson, and D. Spiegelhalter (1995). *Markov Chain Monte Carlo in Practice*. Interdisciplinary Statistics. Boca Raton, FL.: Chapman & Hall / CRC.

Harter, R., K. Hornik, and S. Theussl (2012). *Rsymphony: Symphony in R*. R package version 0.1-14.

Henningsen, A. (2010). *linprog: Linear Programming / Optimization*. R package version 0.9-0.

Konis, K. (2011). *lpSolveAPI: R Interface for lp_solve version 5.5.2.0*. R package version 5.5.2.0-5.

Maillard, S., T. Roncalli, and J. Teiletche (2009, May). On the properties of equally-weighted risk contributions portfolios. Working paper, SGAM Alternative Investments and Lombard Odier and University of Paris Dauphine.

Maillard, S., T. Roncalli, and J. Teiletche (2010). The properties of equally weighted risk contribution portfolios. *The Journal of Portfolio Management 36*(4), 60–70.

Markowitz, H. (1952, March). Portfolio selection. *The Journal of Finance 7*(1), 77–91.

Marschinski, R., P. Rossi, M. Tavoni, and F. Cocco (2007). Portfolio selection with probabilistic utility. *Annals of Operations Research 151*, 223–239.

Bibliography III

Michaud, R. (1989). The markowitz optimization enigma: Is optimized optimal. *Financial Analyst Journal 45*, 31–42.

Michaud, R. (1998). *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*. New York: Oxford University Press.

Mullen, K., D. Ardia, D. Gil, D. Windover, and J. Cline (2011). DEoptim: An R package for global optimization by differential evolution. *Journal of Statistical Software 40*(6), 1–26.

Neal, R. (2011). *Handbook of Markov Chain Monte Carlo*, Chapter MCMC using Hamiltonian dynamics, pp. 113–162. Handbooks of Modern Statistical Methods. Boca Raton, FL: Chapman & Hall / CRC.

Novomestky, F. (2012). *rportfolios: Random portfolio generation*. R package version 1.0.

Peterson, B. and K. Boudt (2008, November). Component var for a non-normal world. *Risk*. Reprint in Asia Risk.

Pfaff, B. (2011). *rneos: XML-RPC Interface to NEOS*. R package version 0.2-6.

Pfaff, B. (2012). *Financial Risk Modelling and Portfolio Optimisation with R*. London: John Wiley & Sons, Ltd.

Qian, E. (2005). Risk parity portfolios: Efficient portfolios through true diversification. White paper, PanAgora, Bostan, MA.

Qian, E. (2006). On the financial interpretation of risk contribution: Risk budgets do add up. *Journal of Investment Management 4*(4), 1–11.

Qian, E. (2011, Spring). Risk parity and diversification. *The Journal of Investing 20*(1), 119–127.

Rossi, P., M. Tavoni, F. Cocco, and R. Marschinski (2002, November). Portfolio selection with probabilistic utility, bayesian statistics and markov chain monte carlo. *eprint arXiv arXiv:cond-mat/0211480*, 1–27. http://arxiv.org.

Rowe, B. (2012). *tawny: Provides various portfolio optimization strategies including random matrix theory and shrinkage estimators*. R package version 2.0.2.

Schaible, S. (1967a). Fractional programming. i, duality. *Management Science 22*(8), 858–867.

Schaible, S. (1967b). Fractional programming. ii, on dinkelbach's algorithm. *Management Science 22*(8), 868–873.

Soetaert, K., K. Van den Meersche, and D. van Oevelen (2009). *limSolve: Solving Linear Inverse Models*. R package 1.5.1.

Stoyanov, S., S. Rachev, and F. Fabozzi (2007). Optimal financial portfolios. *Applied Mathematical Finance 14*(5), 401–436.

Bibliography IV

Theussl, S. and K. Hornik (2012). *Rglpk: R/GNU Linear Programming Kit Interface*. R package version 0.3-8.

Turlach, B. A. and A. Weingessel (2011). *quadprog: Functions to solve Quadratic Programming Problems.* R package version 1.5-4.

Würtz, D., Y. Chalabi, W. Chen, and A. Ellis (2010a, April). *Portfolio Optimization with R/Rmetrics*. Rmetrics Association & Finance Online, www.rmetrics.org. R package version 2130.80.

Würtz, D., Y. Chalabi, W. Chen, and A. Ellis (2010b, April). *Portfolio Optimization with R/Rmetrics*. Rmetrics Association & Finance Online, www.rmetrics.org. R package version 2110.4.

Ye, Y. (1987). *Interior Algorithms for Linear, Quadratic, and Linearly Constrained Non-Linear Programming*. Ph. D. thesis, Department of ESS, Stanford University.