universität wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

"Portfolio Optimization via Smoothing the V@R"

verfasst von / submitted by

Mag. Christoph Schachinger Bakk.rer.soc.oec.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Magister der Sozial- und Wirtschaftswissenschaften (Mag.rer.soc.oec)

Wien, 2018 / Vienna 2018

Studienkennzahl lt. Studienblatt:	A 066 951
Studienrichtung lt. Studienblatt:	Statistik
Betreuer:	o. Univ.-Prof. Mag. Dr. Georg Pflug

Contents

Abbreviations and Mathematical Notation

Abbreviation	
rv	random variable
pdf	probability density function
cdf	cummulative distribution function
a.s.	almost surely
LP	Linear (optimization) Program
NLP	Non-linear Program
V@R	Value-at-Risk
AV@R	Average Value-at-Risk
smV@R	smoothed Value-at-Risk

Notation	
$\rho(\cdot)$	risk measure or functional
$f_X(x)$	pdf of rv X
$F_X(x)$	cdf of rv X
$F_X^{-1}(\alpha)$	quantile function of rv X
$\hat{F}_X(x)$	empirical/estimated cdf of X
$\hat{F}_X^{-1}(\alpha)$	empirical quantile function of rv X
$D_f(\cdot)$	Jacobian matrix of function $f(\cdot)$
n	number of securities
m	number of scenarios
i	security ID (i = 1,. . . ,n)
j	Scenario ID (j = 1,. . . ,m)
ξ	random vector of returns
$\xi_{j,i}$	return of security i in scenario j (realization of ξ)
$\xi_{\cdot,\cdot}$	full data matrix of returns (multiple realizations of ξ)
$\bar{\xi}$	mean return vector for all assets (i = 1,. . . ,n)
x_i	optimization variable ($i \in 1, ..., n$)
$g(x, \xi)$	return function
θ	smV@R value in optimization
α	probability for risk functional
$\mathbb{1}(A)$	indicator function (1 if $x \in A$ else 0)
$\phi(A)$	smoothing function of indicator function
$\lfloor \beta \rfloor$	largest $z \in \mathbb{Z} : z \leq \beta, \beta \in \mathbb{R}$
σ_X	standard deviation of rv X
$\sigma_{X,Y}$	covariance of rvs X and Y
Σ	Variance-Covariance matrix

Acknowledgements

In the beginning I would like to thank everybody who supported me with advice - topicrelated or not - and discussions, supportive words or an open ear when needed. Especially my supervisor, Professor Georg Pflug, helped me understand the caveats and problems in the area of portfolio selection and stochastic programming. Without these insights this work would not have been possible in the presented form.

Additionally, I would like to give special thanks to my parents with their encouragement and efforts of cheering me up.

Introduction

In this thesis the topic of portfolio selection will be investigated upon and several approaches will be discussed and compared.

In the first section the setup will be presented accompanied by a short discussion on the currently most relevant approaches.

The next section covers and defines some attributes of risk measures and the later used risk measures are introduced. Since there already exist some standard methods for optimization approaches based on the AV@R and the standard deviation as risk criterion they will be used to compare and benchmark the smoothing approach to the V@R. For the latter - the V@R - no standard methods have become widely accepted yet and made it necessary to come up with some new ideas or refine and apply existing ones. The result - the smoothed V@R in the presented form - is the subject of the next section. In the first steps of this section a promising formulation for the optimization is developed and introduced in detail. The core of this approach is to approximate the (empirical) cdf by a twice continously differentiable function that also is flexible when it comes to the choice of the mollifier - i.e. the approximation of the indicator function. Additionally their (first) derivate is given as closed form function and some challenges are discussed.

The final section is dedicated to show the applicability of the developed approach to a random and a real world example. This part of the thesis also covers - in part - the intricacies of the choice of non-linear optimization algorithms and corresponding parameters. After an adequate NLP algorithm is found the results of the smoothed V@R optimization is compared with the standard methods based on the standard deviation and the AV@R.

The conclusion finalizes this thesis and completes it with reflections on opportunities and ideas for further refinements of the presented method of the smoothed V@R.

Setup and Definitions

When it comes to portfolio selection one is confronted with several difficulties to tackle. To outline some the below - not necessarily complete - list should give a short overview.

- Utility Function(s)
- Definition of Optimality
- Limitations and Costs of Trading or Contracting
- Generalizability of Conclusions
- Representativeness
- Mathematical Properties and available Models
- Modelling Contingencies

More generally speaking, the framework examining those questions is "Stochastic Programming". Due to the difficulties stemming from the operationalization of general utility functions, their random and mathmatical properties as well as the lack of intuitive interpretation make such general approaches inopportune and bear the risk of high effort when the approach has to be adapted to changed circumstances. For that reason the subsection Mean-Risk modelling has been developed. It is already rather well explored and major applications in the finance industry are based on it.

In the Mean-Risk models one is confronted with the problem of distributing the available ressources optimally using the trade-off between the mean/expected return and some risk measure. To be more explicit there are - mainly equivalent - approaches to either minimize the risk measure at hand while specifying a lower bound for the mean return or to maximize the return/profit of available financial instruments constrained by a maximum risk exposure. This approach is considered to be a rather risk averse approach.

Concerning the risk measures an extensive bandwith is available. The below list gives some insight in the main risk measures currently applied:

- Standard Deviation/Variance/Volatility/Covariance matrix
- Semideviations
- Value-at-Risk (henceforth V@R)
- Weighted Mean Deviations from Quantiles
- Average/Conditional Value-at-Risk/Expected Shortfall/Expected Tail Loss (henceforth AV@R)
- Entropic Value-at-Risk (EV@R)

The previously mentioned risk measures and their properties are the subject of a considerable amount of literature and to cover all of them satisfatorily would stretch far beyond the focus of this thesis. Therefore only a selection will be characterized and formally introduced - namely the standard deviation, the V@R and the AV@R.

The selection of the standard deviation is mainly driven by the observation that it is the risk measure that drove the development of the Capital Asset Pricing Model (henceforth CAPM) and is historically the first major approach in portfolio optimization. Additionally it can be found as the core element of a surprisingly large number of software implementations for portfolio selection up until today.

The risk measures V@R and AV@R are a more obvious choice since they are closely related and the modern standard for portfolio selection, the CAPM and diverse regulatory requirements in the finance industry (e.g. capital requirements for insurances and banks - see Solvency II and Basel

III + IV). While the AV@R is currently viewed as the "Gold Standard" due to its advantagous mathematical and risk capturing properties (e.g. coherence, convexity, $V@R \leq CV@R$, AV@R also covers "tail risk"). Nevertheless the V@R is still the central risk measure of regulatory choice and is more intuitive to understand.

For a formal investigation and description some definitions have to be made. The following section is dedicated to this.

Formal Setup

Portfolio Mix

Conventionally the portfolio mix is characterized by the letter x and assumed to be element of $\mathbb{R}^n$. Therefore it is a vector with components x_i with $i \in \{1, ..., n\}$ giving the portion of asset i in the selected portfolio consisting of n possible investments. For sake of simplicity they are assumed to be assigned proportions of the starting capital so that they sum up to 1 and can be applied irrespective of e.g. the starting capital.

Random Returns

The random returns will be refered to as ξ - a random vector of dimension n. The realizations of that random vector or its components will be denoted in matrix form with entries $\xi_{j,i}$ where j is the ID of the scenario and i the ID of the asset. Since it is necessary to strictly differentiate between random vectors and their realizations - i.e. in form of a data set - the complete matrix of realizations will be referred to as $\xi_{.,.}$. For simplicity and neglecting possible other random impacts (e.g. conditional references to economic conditions, random costs from transactions etc.) ξ is assumed to be the only source of randomness in the presented model.

Mean Return

Despite the fact that the below approach could be applicable to non-linear portfolio return functions $g(x, \xi)$ this work is centered around the linear return function - namely $g(x, \xi) = \xi' x$. So the mean return is $E(g(x, \xi)) = E(\xi)' x$ and $E(\xi)$ will be replaced with the vector $\bar{\xi}$ containing the arithmetic mean of each asset i as components - a consistent estimator if the expected returns exist (and are finite).

Characterization of Return

Alongside the choice of the risk measure the mathematical characterization of the random vector ξ can be viewed as the second core modelling decision and is essential - despite the fact that it is often not mentioned explicitly. This decision dictates - to a material extent - the possibilities of the subsequent optimization model. As an example the prominently applied assumption of (Gaussian) normal distribution (or other Eliptic distributions) of returns (or e.g. log(return)) should be mentioned where analytical solutions are available in many areas. More generally the - at least partially - arbitrary assumptions of analytically well examined distributions (e.g. log-normal) or

copulas also have to be cited here. While the assumption of a closed form distribution offers efficient models the question of adequacy (e.g. representativeness and generalizability) is often hard - if even possible - to answer. More importantly the data is usually only used to estimate the parameters of the chosen distribution and not utilized in subsequent analyses or inference.

Next to these aforementioned analytical approaches the option of semi- or non-parametric random sampling either from closed form distributions or data (e.g. time series of returns) are frequently used. Data sampling approaches have (much) weaker apriori assumptions but are doomed to be computationally expensive. Additionally the application of the drawn conclusions to future returns remains a question of experience, circumstances and - last but not least - stable behaviour over time of the securities under investigation. Some of the challenges can be included into the model but they limit the generalizability.

In spite of the interesting questions raised beforehand they are - to a large extent - out of scope of this thesis and have to be subjected to the modelling decision made in this thesis. Two different choices have been made. For illustrative purposes of the smoothed V@R approach, data is sampled from a closed form (and well-defined) distribution and in the final section random sampling from real time series data is used to find an optimal portfolio.

Risk Measures

When it comes to portfolio selection and risk measures several attributes of those measures have been developed to analyse and anticipate the behaviour of said measures. Whilst some of those attributes are deemed advantagous others are necessary for interpretation and comparison of portfolios. Since the selected risk measures are already well examined this section is more dedicated to the later discussion of the applied risk measures than to provide strict mathematical proofs or analyses.

Returns versus Losses

It has to be pointed out that in most of the literature about risk measures or functionals the definitions of the later mentioned attributes are for **loss** random variables. Since in this thesis **returns** are used to model the random outcomes of investment decisions some attributes have to be slightly adapted to be consistent with these definitions. To avoid confusion **both** definitions are given in table form if adaptions are necessary. The following definition should help to capture the difference where υ denotes losses and - as already mentioned - ξ denotes returns - and therefore $\upsilon := 1 - \xi$. If the property **does not depend** on whether losses or returns are used the random outcome is denoted by X.

Essential Properties

For a consistent estimation of the expectation of a random variable it is imperative that the expected value is finite. Owing to the fact that also the variance/standard deviation will be used to measure risk also the second moments of the underlying distribution of returns has to be finite. In a more formal framework this means that the returns have to fulfill at least $\xi \in \mathcal{L}_2(\Omega, \mathcal{F}, \mathcal{P})$. Heuristically speaking this means (or implies) that ξ is measurable (in the sense of measure theory) with respect to the σ-algebra $\mathcal{F}$ over Ω by the measure $\mathcal{P}$ and $E(\xi^2) < \infty$. More plainly speaking these properties

are a general prerequisite to use the calculus of probability in the following sections and to apply the risk measures mentioned later.

Advantagous Properties

Whereas the previous essential properties would principally prohibit a meaningful portfolio selection the following properties are deemed "only" advantagous. "Advantagous" has to be understood that use, analysis and comparison of the risk measures is clearly defined and - with the necessary experience - intuitive (or possibly even automatable). The definitions of the properties are extracted from Pflug/Römisch (2007). In this thesis the terms "risk measure" or "risk functional" are used and will be denoted by $\rho(\cdot)$.

Translation Properties

Translation Equivariance

For all scalars $a \in \mathbb{R}$ and a random outcome X the equality $\rho(X + a) = \rho(X) + a$ holds.

Translation Invariance

For all scalars $a \in \mathbb{R}$ and a random outcome X the equality $\rho(X + a) = \rho(X)$ holds.

Translation Antivariance

For all scalars $a \in \mathbb{R}$ and a random outcome X the equality $\rho(X + a) = \rho(X) - a$ holds.

Monotonicity

Pointwise Monotonicity

If the outcome of a random event is almost surely (henceforth a.s.) worse than another, the risk functional strictly captures that. Formally that means:

RETURNS ξ	**LOSSES** υ
If $\xi_1 \leq \xi_2$ a.s. then $\rho(\xi_1) \leq \rho(\xi_2)$	If $\upsilon_1 \leq \upsilon_2$ a.s. then $\rho(\upsilon_1) \geq \rho(\upsilon_2)$

The above equivalence can be seen easily since $\xi_1 \leq \xi_2 \Leftrightarrow \upsilon_1 \geq \upsilon_2$.

Scaling Properties

Homogeneity

A risk functional is called homogenous if for a scalar $\lambda \in \mathbb{R}$ and a random outcome X it holds that $\rho(\lambda X) = \lambda \rho(X)$.

Positive Homogeneity

A slightly weaker version of homogeneity is positive homogeneity. It is fulfilled if for a scalar $\lambda \in \mathbb{R}^+$ and a random outcome X it holds $\rho(\lambda X) = \lambda\rho(X)$. In other words the risk of a position grows relative to its size.

Additivity

Additivity

The risk functional of a position X must be equal to the sum of the risk of its constituent parts - i.e. $\rho(X_1 + X_2) = \rho(X_1) + \rho(X_2)$.

Subadditivity

The risk functional of a position X may not be higher than the sum of the risk of its constituent parts - i.e. $\rho(X_1 + X_2) \leq \rho(X_1) + \rho(X_2)$.

Superadditivity

The risk functional of a position X may not be lower than the sum of the risk of its constituent parts - i.e. $\rho(X_1 + X_2) \geq \rho(X_1) + \rho(X_2)$.

Shape Properties

Convexity

This property has no direct intuitive interpretation but captures very advantagous optimization attributes. For all scalars $\lambda \in [0, 1]$ $\rho(\lambda X_1 + (1 - \lambda)X_2) \leq \lambda\rho(X_1) + (1 - \lambda)\rho(X_2)$ holds.

Concavity

For all scalars $\lambda \in [0, 1]$ $\rho(\lambda X_1 + (1 - \lambda)X_2) \geq \lambda\rho(X_1) + (1 - \lambda)\rho(X_2)$ holds.

Coherence

This attribute is a bundle of properties. A risk measure $\rho(\cdot)$ is called *coherent*[1] iff it is *pointwise antimonotone*, *translation antivariant*, *subadditive* and *positive homogenous*. Another definition requires that the risk measure must be *convex*. Since a risk measure is convex iff it is subadditive and positive homogeneous these definitions are equivalent.

[1] Pflug/Römisch (2007, p. 38ff) and Shapiro et al. (2009 p. 261)

Standard Deviation

One of the most prominent examples of measures for the "width", "stretch" or dispersion of a random distribution is the standard deviation. As the square root of the variance in the following sections the terms "variance" and "standard deviation" are used as equivalent risk measures in the sense that translation from one into the other is very simple and since $\sigma \geq 0$ it is also bijective. The variance is symbolized and defined by $\sigma^2 = E[(X - E(X))^2]$ and consistently estimated by $\hat{\sigma}^2 = s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2]$ - with $\bar{x}$ as the sample aritmethic mean.

In a multivariate setting the covariance matrix Σ can be viewed as the multidimensional equivalent of σ^2 also capturing the covariances (e.g. rotation in a multivariate normal setting) in the off-diagonal elements of the matrix.

$\Sigma = E[(X - E(X))(X - E(X))'] = (\sigma_{i,j}) \in \mathbb{R}^{n \times n}$

This covariance matrix is the basis for the optimization approach of Markowitz since the variance of the final portfolio can be derived from it and thus easily approximated via estimates.

Mathematical Properties

The standard deviation σ

- does not differentiate between up- and downwards deviations from the mean - so it is a mean-symmetric measure (i.e. positive and negative return deviations from the mean are penalized equally if their absolute value is equal)
- is a **deviation risk measure**
- today it is still applied *like* a risk measure in multiple portfolio selection procedures
- is positive homogenous ($\sigma_{aX} = a\sigma_X$)
- is **NOT** monotone (let $a \in (0, 1)$, $X_1 = a \cdot X_2$ and $X_2 \sim U[0, 1]$ then $X_1 \leq X_2$ a.s. but also $\sigma^2_{X_1} = a^2 \sigma^2_{X_2} \leq \sigma^2_{X_2}$)
- is translation invariant ($\sigma^2_{X+a} = \sigma^2_X$) and thus neither translation equi- or antivariant
- fulfills **NONE** of the above additivity properties ($\sigma_{X_1+X_2} = \sqrt{\sigma^2_{X_2} + \sigma^2_{X_2} + 2 \cdot \sigma_{X_1,X_2}}$)

The covariance matrix Σ

- is a quadratic, symmetric and non-negative definite matrix ($\Sigma' x \geq 0$ for all x)
- is simple to estimate
- estimates can be shown - under rather loose conditions - to converge to the real parameter matrix (if the latter is well defined)
- is directly usable in a quadratic optimization model which is convex (so solutions to such an optimization are unique if the problem is feasible)
- is equivalent with V@R and AV@R optimization for Normally distributed returns (with proper matrix Σ)
- captures only linear dependencies between components of the random vetor X (through covariance)

Short discussion

Since up- and downwards deviations are not accounted for differently the standard deviation may not be an appropriate measure. Especially in settings of skewed return distributions the standard deviation over- or underestimates the risk to capture. The discrimination of up- and downwards deviations motivated several other deviation risk functionals like e.g. the lower semi standard deviation $\sigma^- = \sqrt{(E[X - E(X)]^-)^2}$ only accounting for negative deviations from the mean[2].

Despite the above drawbacks it is still a valid benchmarking model with a good balance between simplicity and focus on important features of distributions compared to the discussed disadvantages.

Optimization Formulation

The optimization formulation goes back to Markowitz (1952) and is a quadratic optimization problem. Such optimization problems have the following general structure:

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{min} - d'x + \frac{1}{2} x'Dx \\ &\text{s.t. } A'x \geq b_0 \end{aligned}$$

Substituting the matrix Σ, the mean returns $\bar{\xi}$ plus the minimum desired return μ_{min} and the requirement $\mathbb{1}'x \geq 1$ for the corresponding variables the simple quadratic program to solve is given by:

$$\begin{aligned} &\underset{x \in \mathbb{R}^{n+}}{min}\ x'\Sigma x \\ &\quad \text{s.t. } \bar{\xi}'x \geq \mu_{min} \\ &\qquad \mathbb{1}'x \geq 1 \end{aligned}$$

This optimization problem can be solved efficiently and in high dimensions (large n). It is noteworthy that in this optimization problem only aggregated (and estimated) parameters are utilized in the optimization. This makes it powerful in high dimensional data but all other characteristics of the data set are not taken into account.

Value-at-Risk

Probably the most intuitive Risk measure is the Value-at-Risk (V@R). Heuristically speaking lower returns or higher losses won't occur with probability $\alpha \in (0, 1)$. As one can see the V@R is a lower or an upper bound depending on if losses or returns are under investigation. Nevertheless the focus is always the quantile of the probability distribution of the loss or the return. To account for this difference and keep the interpretation consistent two definitions - one for returns Y and one for losses L - are given below:

[2] Pflug/Römisch (2007, p. 24)

V@R FOR RETURNS Y

$$V@R_\alpha(Y) = Q_Y(1-\alpha) := \inf_{\theta\in\mathbb{R}}\{\theta : F_Y(\theta) \geq 1-\alpha\} = \inf_{\theta\in\mathbb{R}}\{\theta : P(Y \leq \theta) \geq 1-\alpha\}$$

V@R FOR LOSSES L

$$V@R_\alpha(L) = Q_L(\alpha) := \inf_{\theta\in\mathbb{R}}\{\theta : F_L(\theta) \geq \alpha\} = \inf_{\theta\in\mathbb{R}}\{\theta : P(L \leq \theta) \geq \alpha\}$$

Its (discrete) sample equivalents are the $\lfloor m(1-\alpha)\rfloor^{th}$ smallest observed return[3] or the $(\lfloor m\alpha\rfloor + 1)^{th}$ highest loss.

Mathematical Properties

The V@R is **not** a coherent risk measure since it is not subadditive. It is positive homogenous, pointwise monotonous and translation equivariant. For eliptical distributions - like the Gaussian Normal distribution - using the Markowitz approach (standard deviation) yields equivalent optimization results to a V@R optimization.

Short discussion

Using the V@R as risk measure has attracted a lot of criticism due to several aspects of its properties and its application. Especially the missing sub-additivity makes it hard to assess the impact of changes in the portfolio composition. More specifically one has to re-evaluate the distribution or associated parameters for all trading options at hand which renders it inopportune in many cases. Additionally the non-convexity brings the problem of mutiple local - or even global - optima that can be a tough challenge in non-linear programming. An other significant property is that the V@R ignores the so called tail-risk. In other words the magnitude and probability of losses exeeding the V@R is not accounted for. Therefore losses in this so called tail may differ significantly for different portfolios although the V@R is equal[4].

Optimization Formulation

The optimization formulation is a little bit more intricate and an approximative smoother approach will be presented in the section "The Smoothed V@R Approach".

[3] $\lfloor\beta\rfloor$ for $\beta \in \mathbb{R}$ is the largest integer not exeeding β

[4] One simple example would be $P(return_1 = 1) = P(return_2 = 1) = 0.99$ and $P(return_1 = 0.95) = 0.01$ while $P(return_2 = 0.9) = 0.01$. The $V@R_\alpha$ for $\alpha < 0.99$ is the same in both portfolios but the loss would be half for $portfolio_1$ compared to $portfolio_2$.

Average Value-at-Risk

The current Gold standard in Mean-Risk Optimization is the Average Value-at-risk[5]. It can be described as the average loss above the V@R. Since the applied optimization formulation only uses losses the definition below is also given **only for losses**.

Let L be losses (they are negative if in a scenario profit would be made) and $F_L(l)$ their cdf. If $F_L(l)$ is continous at $l^* = V@R_\alpha(L)$ then

$$\begin{aligned} AV@R_\alpha(L) &= (1-\alpha)^{-1} \int_\alpha^1 F_L^{-1}(\tau) d\tau \\ &= (1-\alpha)^{-1} \int_\alpha^1 V@R_\tau(L) d\tau \\ &= (1-\alpha)^{-1} \int_{V@R_\alpha(L)}^{+\infty} l dF_L(l) \\ &= V@R_\alpha(L) + (1-\alpha)^{-1} \int_{l \in \mathbb{R}} [l - V@R_\alpha(L)]^+ dF_L(l) \end{aligned}$$

More details and proofs of the above equations can be found in Pflug/Römisch (2007, p. 48f) and Shapiro et al. (2009, p. 254-261, especially Theorem 6.2 with equations 6.26 and 6.27). In this thesis the optimization formulation of Krokhmal et al. (2001) will be used.

Mathematical Properties

First of all it is a coherent risk measure that is convex and also accounts for the tail risk. Krokhmal et al. (2001) also developed a close approximation of the AV@R for purposes of optimization with transaction costs and other restrictions. This approximation can be used to solve the portfolio optimization problem with Linear Programming tools or software[6]. This formulation of the AV@R problem without any restrictions on the portfolio composition, transactions or transaction costs was used in the comparison of risk measures. It is worthwhile mentioning that the output for the V@R and AV@R in this model is only correct if the AV@R condition is binding. Despite the fact that without any other restrictions this has to be the case in the used application the V@R and AV@R values were computed separately based on the optimal portfolio composition.

Since the AV@R is a convex approximation of the V@R constraint it is often the case that the optimal portfolio for the AV@R is not the same as for the V@R - especially of course if the optimal x for the AV@R is driven more by the tail risk than by the quantile. More heuristically speaking the optimal portfolio composition based on the V@R may - or most likely will - differ from optimal portfolio composition based on the AV@R. The question if that is a desired feature or not should be answered separately based on the objective of the optimization at hand.

[5] also called expected shortfall or average tail risk, the sometimes used term "Conditional Value-at-Risk" or CV@R should be avoided since in probability theory the concept "conditional" is used in a different context

[6] The approximation is not exact since it neglects the probability of the "V@R-scenario" that is non-zero for discrete approximations. The effect is that ζ is not the appropriate V@R for the applied choice of α.

Short discussion

As already mentioned the V@R is the key figure for regulatory purposes in insurances (Solvency II) and banks (Basel III + IV). Owing to the fact that the $V@R_\alpha \leq AV@R_\alpha$ and the simpler handling of the AV@R this conservative approximation is often used to obtain an upper bound for capital requirements originally associated with the V@R. Until said regulatory frameworks shift the paradigm from V@R to AV@R it is very likely that tools simplifying the handling of the V@R are opportune - despite the AV@R being more advantagous in many instances.

Optimization Formulation

In Krokhmal et al. (2001) a full linear optimization program (henceforth LP) is developed that also incorporates transaction costs as well as other portfolio or transaction restrictions. Since they are no desired feature in this thesis they where left out. Taking the last formulation of the AV@R from above and replacing the $V@R_\alpha(L)$ by ζ the AV@R can be approximated by a discrete weighted sum[7]:

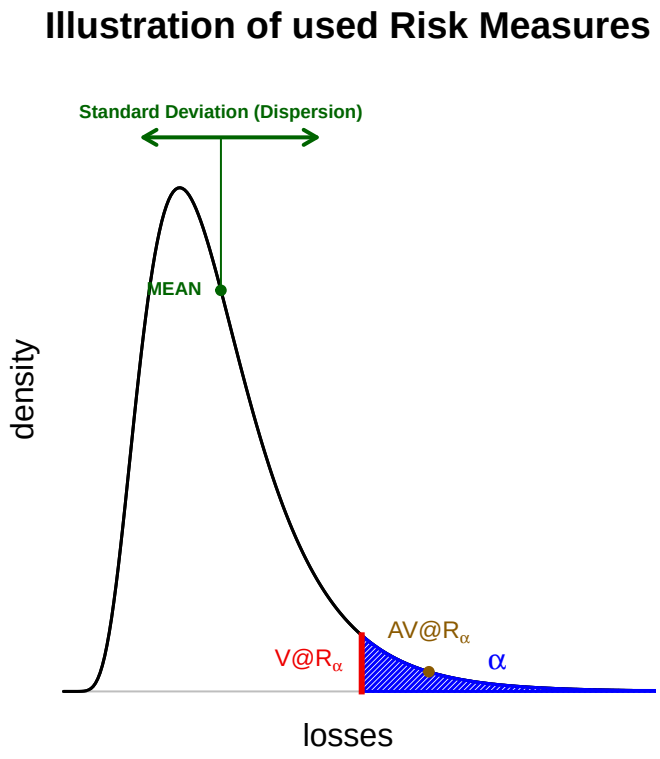
$$\zeta + (1-\alpha)^{-1} \int_{l \in \mathbb{R}} [l-\zeta]^+ dF_L(l) \approx \zeta + (1-\alpha)^{-1} \sum_{j=1}^{m} \pi_j [l_j - \zeta]^+$$

where π_j is the probability of scenario j and of course $\sum_{j=1}^{m} \pi_j = 1$. In the setting at hand all π_j are equal - meaning $\pi_j = \pi = \frac{1}{m}, \forall j$. After replacing $[l_j - \zeta]^+$ with a support variable r_j and providing a maximal exposure value ω_{max} the AV@R constraint can be used in the following LP:

$$\begin{aligned}
\min_{(x,\zeta) \in \mathbb{R}^{n+} \times \mathbb{R}} \quad & -\bar{\xi}' x & \\
\text{s.t. } & \zeta + (1-\alpha)^{-1} \sum_{j=1}^{m} \pi r_j \leq \omega_{max} & j = 1, ..., m \\
& r_j \geq (1 - \sum_{i=1}^{n} \xi_{j,i} x_i) - \zeta & j = 1, ..., m \\
& r_j \geq 0 & j = 1, ..., m \\
& \mathbb{1}' x \leq 1 &
\end{aligned}$$

This program can be solved very efficiently and will be the second benchmark for the smoothed V@R approach that will be presented in the next section.

[7] Krokhmal et al. (2001, p. 6 and 9)



The Smoothed V@R Approach

General Discussion

In an outlining presentation Pflug (2005b) offered two main approaches to optimize a portfolio using the V@R as risk functional:

- Censoring (e.g. AV@R)
- Smoothing (Gaivoronski/Pflug 2005)

In another article Gaivoronski/Pflug (2005) developed one version of such a smoothed V@R. The central idea was to obtain an approximation of the V@R that is twice continously differentiable and thus has good properties for efficient non-linear optimization algorithms. To achieve this goal an ordering algorithm was developed that took into account that information on distributions of securities usually is only available in implicit and discrete form[8]. Inspired by this smoothing approach another alternative to Gaivoronski/Pflug (2005) is outlined in the next sections that aims at a simplification of the idea while maintaining twice continous differentiability.

It should be mentioned that Shapiro et. al. (2009, p. 257f) also use a smoothing approach when introducing the AV@R. By applying a smoothing function to the V@R they outline that the AV@R can be formulated as the result of a smoothing approach as well.

[8]The cdf of data about returns or losses is usually not explicitly known and either has to be estimated based on discrete time series data or "classical" distribution assumptions are made that guide the estimation of the characterizing parameters.

Problem definition

The main goal is to obtain a non-linear program (henceforth NLP) of the form:

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{min} f(x) \\ &\text{s.t. } g(x) = 0 \\ &\quad lb \leq h(x) \leq ub \\ &\quad l_x \leq x \leq u_x \end{aligned}$$

Favorable properties of the functions $f(x)$, $g(x)$ and $h(x)$ would be that they are either linear, quadratic, convex or at least twice continously differentiable since such NLPs can be tackled with rather efficient algorithms.

Used Formulation of Portfolio Selection Problem

The problem of Portfolio selection based on the risk measure V@R will furthermore be formulated in the following way:

$$\begin{aligned} &\underset{x\in\mathbb{R}^{n+}}{min} -E(g(x,\xi)) \\ &\text{s.t. } V@R_\alpha(g(x,\xi)) \geq \theta_{min} \\ &\quad \mathbb{1}'x = 1 \end{aligned}$$

It has to be stressed here that since $g(x,\xi)$ characterizes the returns also the V@R will be **used in the form defined for returns**.

After some formal changes, replacing the random variable ξ with its realizations $\xi_{j,i}$ and the functionals by its sample equivalents[9] as well as fixing $g(x,\xi) = \xi'x$ - so a linear return function - we obtain the following intermediate formulation:

$$\begin{aligned} &\underset{x\in\mathbb{R}^{n+}}{min} -\bar{\xi}'x \\ &\text{s.t. } \hat{F}^{-1}_{\xi'x}(1-\alpha) \geq \theta_{min} \\ &\quad \mathbb{1}'x = 1 \end{aligned}$$

$\hat{F}^{-1}_{\xi'x}(1-\alpha)$ is not a very nice function since it involves the inverse of a counting function but with some changes one can find an equivalent formulation. We can use the definition of the quantile function $F^{-1}(1-\alpha) := inf(\theta : F(\theta) \geq 1-\alpha)$ to replace this problematic function. To come up with a usable form we need some more thoughts. Let the returns $Y^{(x)} = \xi'x$ then their empirical cdf can be written as

[9] $\frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n}\xi_{j,i}x_i = \frac{1}{m}\sum_{j=1}^{m}\xi'_{j,\cdot}x = \bar{\xi}'x$ as the estimator of $E(g(x,\xi))$

$$\hat{F}_{Y^{(x)}}(\theta) := \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}(y_j^{(x)} \leq \theta) = \frac{1}{m}\sum_{j=1}^{m} \mathbb{1}(y_j^{(x)} - \theta \leq 0)$$

This can be used as a constraint to obtain the inverse $\hat{F}^{-1}_{Y^{(x)}}(1-\alpha)$ by solving for $inf(\theta : F_{Y^{(x)}}(\theta) \geq 1-\alpha)$. The problem is that $F_{Y^{(x)}}$ is a step function and not differentiable. Now the idea is to replace the indicator function $\mathbb{1}(y_j^{(x)} - \theta \leq 0)$ with a twice continously differentiable function $\phi_\epsilon(y_j^{(x)} - \theta)$ that allows to approximate the indicator arbitrarily close via the bandwidth parameter ϵ. Since sum (and multiplication) maintain twice continously differentiability (henceforth tcd) we could have come up with a proper version of a smoothed empirical cdf.

For now let $S(\xi_{\cdot,\cdot}, x, \theta, \epsilon)$ be the smoothed empirical cdf that is not defined in detail yet. If we are interested in a proxy for the V@R for a fixed portfolio composition x we can use this function to define the following program already:

$$\begin{aligned} &\underset{\theta\in\mathbb{R}}{min}\ \theta \\ &\text{s.t. } S(\xi_{\cdot,\cdot}, x, \theta, \epsilon) \geq 1-\alpha \end{aligned} \tag{1}$$

This program will be used in the later sections to show how the shape of the V@R is smoothed. Evidently the function $S(\xi_{\cdot,\cdot}, x, \theta, \epsilon)$ needs some more elaboration to be applicable (see section "The Smoothing Function").

Going back to portfolio selection one can use the constraint $S(\xi_{\cdot,\cdot}, x, \theta, \epsilon) = 1-\alpha$ and the simple additional box constraint $\theta_{min} \leq \theta \leq \theta_{max}$ to obtain the inverted value θ and the corresponding bounds. θ_{min} denotes the maximum V@R allowed ($1-\theta_{min}$ is the maximum loss) while θ_{max} is a moderating constant that should prevent the NLP algorithm from leaving the domain of definition for the function $S(\xi_{\cdot,\cdot}, x, \theta, \epsilon)$ inappropriately far. Depending on starting value, software and applied algorithm the latter value can vary. Experience from the experiments shows that a value of $max(\xi_{j,i})$ should suffice but can be chosen in a different way as well (e.g. if a high α is chosen).

Since the optimum mean value of a portfolio is always lower if not the full available ressources are distributed the formulation $\sum_{i=1}^{n} x_i \leq 1$ is equivalent to $\sum_{i=1}^{n} x_i = 1$ but more advantagous in the implementation. Firstly the result will not be influenced materially and secondly a "line" search is more prone to termination at a local optimum than a "space" search offering more directions for the next step. Additionally NLP algorithms allowing non-linear constraints face the difficulty of what to do if the constraints are violated at an intermediary result. This problem can be mitigated by using inequality instead of equality constraints.

In the last step we want to get rid of the second inner minimization problem caused by the infimum in the definition of the quantile function to avoid a two stage optimization problem. To do that we use that for returns $V@R_\beta \leq V@R_\alpha$ if $\beta \geq \alpha$ (and therefore $1-\beta \leq 1-\alpha$). This can be established by the property that $\hat{F}_{Y^{(x)}}$ is monotonously increasing in θ[10]. Hence if we use $\hat{F}_{Y^{(x)}}(\theta) \leq 1-\alpha$ we can deduce the ineqality $\hat{F}_{Y^{(x)}}(\theta') = 1-\beta \leq 1-\alpha = \hat{F}_{Y^{(x)}}(\theta^*)$ and from that $\theta_{min} \leq \theta' \leq \theta^*$. As a consequence θ is an approximation of the adequate $V@R_\alpha$ that still fulfills the required lower bound θ_{min}. Since the smoothed variant of the empirical cdf $S(\xi_{\cdot,\cdot}, x, \theta, \epsilon)$ is also monotonously increasing

[10] $\hat{F}_{Y^{(x)}}(\theta') \leq \hat{F}_{Y^{(x)}}(\theta^*) \Leftrightarrow \theta' \leq \theta^*$

in θ the same applies here. A second measure to capture the infimum will be presented in the next section (see "Applied Smoothers").

After including the above remarks the optimization problem at hand is given below.

$$\begin{aligned} \underset{(\theta,x)\in\mathbb{R}\times\mathbb{R}^{n+}}{min} \quad & -\bar{\xi}'x \\ \text{s.t. } & S(\xi_{.,.},x,\theta,\epsilon) \coloneqq \frac{1}{m}\sum_{j=1}^{m}\phi_\epsilon(\xi_{j,.}x-\theta) \leq 1-\alpha \\ & \theta_{min} \leq \theta \leq \theta_{max} \\ & \mathbb{1}'x \leq 1 \end{aligned}$$

This simple and efficient formulation gives a very good approximation of the V@R - of course strongly depending on the selected function $\phi_\epsilon(\cdot)$ and ϵ. Apparently the number of observations/scenarios also contributes to the potential quality of the approximation.

The optimization problem above will be used to derive an efficient bound comparing mean return and risk. In such a framework minimizing the risk while maintaining a minimum expected return is equivalent to maximizing the expected return while the risk remains below a certain threshold. Using this efficient bound an optimal portfolio can be chosen. This is mentioned explicitly because of the experience that the first problem setup (min risk s.t. return > threshold) gave unsatisfactory outcomes due to the fact that the portfolio mix x did not vary "enough" in the optimization run and the obtained solutions where concentrated locally around the starting values - i.e. the search for the optimum was local. This is not the case in the outlined and implemented formulation. It is worthwhile mentioning that this is not very surprising since almost all derivative based NLP software implementations (allowing non-linear constraints) are explicitly characterized as beeing "local" optimizers to a certain extent.

The Smoothing Function

Necessary Properties

The smoothing function - denoted by $\phi_\epsilon(\cdot)$ - can be chosen rather freely in the described setup. Only some requirements are deemed necessary:

1. **Twice-continous differentiability** (henceforth tcd) for $\epsilon > 0$: Though this requirement is fulfilled by many functions it is also advisable to use a function that has non-zero derivatives on a rather large domain. Especially if numeric derivatives are used "flat" areas (i.e. the derivative is zero in all directions) can lead to singularities and infinite values and result in erroneous termination, step size or else. If precautions are taken in that respect the performance will be much more reliable and stable. As a reminder it also should be kept in mind that a computer program may have a hard time to discriminate zero from almost zero - e.g. if ϵ is too close to 0.
2. **Monotonicity**: Since counting is a monotonous function also the smoother has to incorporate that property. Monotonously decreasing or increasing functions would only differ minimally in matters of implementation.

3. **Efficient evaluation**: Since the function is evaluated multiple times the evaluation of this function strongly determines the run time and computational effort.
4. Controllable **domain of definition** (via ϵ): The domain of the returns per scenario and the distance between them can vary heavily with the used data as well as with the number of scenarios. It is strongly advised to account for that by beeing able to vary the domain of definition of the smoothing function (see also *twice-continous differentiability*).
5. **Domain of function image** is $[0, 1]$: Since this function has to approximate the indicator the image of the function should be $[0, 1]$ - or at least $[0 + \delta, 1 - \delta]$ for a small $\delta \geq 0$.
6. **Symmetry**: If the function $\phi'_\epsilon(z)$ is symmetric then equivalent upper and lower approximations of the indicator function can be achieved easily. This is not a necessary but an advantagous property.
7. $\phi_0(z) = \mathbb{1}(z \leq 0)$
8. $\phi_\epsilon(z) \to 1$ for $\epsilon \to 0$ for any fixed $z \leq 0$
9. $\phi_\epsilon(z) \to 0$ for $\epsilon \to 0$ for any fixed $z > 0$

Applied Smoothers

As was already mentioned the indicator for the empirical cdf $\mathbb{1}(g(x, \xi_{j,.}) = \xi_{j,.}x \leq \theta) = \mathbb{1}(\xi_{j,.}x - \theta \leq 0)$ will be approximated by $\phi_\epsilon(z_j^{(x,\theta)})$ where $z_j^{(x,\theta)} = \xi_{j,.}x - \theta$. Applying $\phi_\epsilon(z_j^{(x,\theta)})$ to each scenario $j = 1, ..., m$ and taking the mean over all m scenarios we obtain a proper approximation $S(\xi_{.,.}, x, \theta, \epsilon)$ of $\hat{F}_{\xi' x}(\theta)$. More explicitly:

$$\hat{F}_{\xi' x}(\theta) \coloneqq \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}(\xi_{j,.}x \leq \theta) = \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}(\xi_{j,.}x - \theta \leq 0) = \frac{1}{m} \sum_{j=1}^{m} \mathbb{1}(z_j^{(x,\theta)} \leq 0) \tag{2}$$

$$S(\xi_{.,.}, x, \theta, \epsilon) \coloneqq \frac{1}{m} \sum_{j=1}^{m} \phi_\epsilon(\xi_{j,.}x - \theta) = \frac{1}{m} \sum_{j=1}^{m} \phi_\epsilon(z_j^{(x,\theta)}) =: S_\epsilon(\theta, x) \tag{3}$$

For simplification of notation and for a better accentuation of the relevant factors a little freedom is taken to denote $S(\xi_{.,.}, x, \theta, \epsilon)$ also as $S_\epsilon(\theta, x)$ to point out that for this function the input $\xi_{.,.}$ is fixed and only varies in x, θ and ϵ. It is important to stress that $\phi_\epsilon(z_j^{(x,\theta)})$ and $\phi_\epsilon(z)$ are different in the sense that while the first refers to the data equivalent in scenario j the latter is used to talk about the function ϕ_ϵ with arbitrary inputs $z \in \mathbb{R}$.

It is evident that the choice of $\phi_\epsilon(z)$ determines the main properties of the approximation $S_\epsilon(\theta, x)$. If $\phi_\epsilon(z) \geq \mathbb{1}(z \leq 0)$ for all $z \in \mathbb{R}$ then also $S_\epsilon(\theta, x) \geq \hat{F}_{\xi' x}(\theta)$ and an approximation from above can be obtained.

It is worthwhile mentioning that for a bounded function fulfilling $\phi_\epsilon(z) \geq \mathbb{1}(z \leq 0), \forall z \in \mathbb{R}$ an approximation from below can be achieved by taking $1 - \phi_\epsilon(-z)$.

Proof.

$$\phi_\epsilon(z) \geq \mathbb{1}(z \leq 0)$$
$$\phi_\epsilon(-z) \geq \mathbb{1}(-z \leq 0)$$
$$-\phi_\epsilon(-z) \leq -\mathbb{1}(-z \leq 0)$$

$$1 - \phi_\epsilon(-z) \leq 1 - \mathbb{1}(-z \leq 0) = \mathbb{1}(-z > 0) = \mathbb{1}(z < 0)$$

□

Gaivoronski/Pflug (2005, p. 23) propose a cubic function to approximate the indicator function. It fulfills all attributes defined above (proofs given in Gaivoronski/Pflug (2005)).

$$\phi_\epsilon^{GP}(z) = \begin{cases} 1 & z \leq 0 \\ 1 - \frac{16}{3\epsilon^3} z^3 & z \in (0, \frac{\epsilon}{4}] \\ \frac{5}{6} + \frac{2}{\epsilon} z - \frac{8}{\epsilon^2} z^2 + \frac{16}{3\epsilon^3} z^3 & z \in (\frac{\epsilon}{4}, \frac{3\epsilon}{4}] \\ \frac{16}{3} - \frac{16}{\epsilon} z + \frac{16}{\epsilon^2} z^2 - \frac{16}{3\epsilon^3} z^3 & z \in (\frac{3\epsilon}{4}, \epsilon] \\ 0 & else \end{cases} \tag{4}$$

In several experiments the performance and applicability looked rather good but the function has some drawbacks. While, for example, the tcd condition fully applys the derivatives (first and second) are 0 for all $z \notin (0, \epsilon)$ by (desired) design. Secondly the evaluation of this function - though fast - can be improved by using functions that are implemented generically in R through C++ - needless to say that for other software this may not be applicable.

When thinking about efficiency, (controllable) domain of definition as well as image and simple implementation continous cdfs come to mind. A very well investigated and ubiquitous distribution with rather simply controllable and advantagous features is the normal distribution. Using the normal cdf also has the big advantage that it is implemented very efficiently in R and - at least theoretically - covers the whole space $\mathbb{R}$ with non-zero values as does its derivative the normal pdf. An obvious disadvantage is that - despite the fact that the image covers $(0, 1)$ - for finite values of z and $\sigma > 0$ the values 1 and 0 are never attained. Therefore the strict majorant property $\phi_\epsilon(z) \geq \mathbb{1}(z \leq 0)$ is lost. Anyway the advantages seem to outweigh the drawbacks.

Conventionally the standard normal cdf is denoted by $\Phi_X(x)$ for a random variable $X \sim N(0, 1)$. Any other normally distributed variable $Y \sim N(\mu, \sigma^2)$ can be derived from X via $Y = \mu + \sigma X$ or re-standardized by $X = \frac{Y - \mu}{\sigma}$. It also comes handy that the normal cdf is symmetric - i.e. $1 - \Phi(x) = \Phi(-x)$. To obtain an appropriate smoother we need some mathematical deductions.

The desired smoother $\phi_\epsilon^{norm}(z)$ should have a value of 1 (or close to 1) $\forall z \leq 0$, a value $p \in (0, 1)$ for $z \in (0, \delta(\epsilon))$ and 0 (or close to 0) $\forall z \geq \delta(\epsilon)$. By setting $\Phi(y) = p$ and applying the always properly defined inverse we trivially see that $\Phi^{-1}(p) = y$. Adding $-z$ gives $-z + \Phi^{-1}(p) = -z + y$ and applying the standard normal cdf again we obtain $\Phi(-z + \Phi^{-1}(p)) = \Phi(-z + y)$. This could be already used as a smoother with $\epsilon = 1$ but we want to control the domain by varying ϵ. This can be achieved by dividing z by ϵ - i.e. $\Phi(-\frac{z}{\epsilon} + \Phi^{-1}(p)) = \Phi(-\frac{z}{\epsilon} + y)$. Through varying y we can attain any value in $(0, 1)$ but $\Phi^{-1}(p)$ directly determines the attained value through p. If we set $\Phi(-\frac{z}{\epsilon} + \Phi^{-1}(p)) = \phi_\epsilon^{norm}(z)$ the main properties can be established very easily.

$$\text{Let } \phi_\epsilon^{norm}(z) \coloneqq \Phi(-\frac{z}{\epsilon} + \Phi^{-1}(p)), z \in \mathbb{R} \text{ (Definition)} \tag{5}$$

$$\phi_\epsilon^{norm}(z) < \phi_\epsilon^{norm}(z') \iff z > z' \text{ and } \epsilon > 0 \text{ (Strict monotonically decreasing)} \tag{6}$$

Proof. Follows from $\Phi(x)$ beeing strictly monotonically increasing and $-\frac{z}{\epsilon} + \Phi^{-1}(p) = x(z) < x(z')$. □

$$0 < p < 1, \text{ and } -\frac{z}{\epsilon} + \Phi^{-1}(p) \in \mathbb{R} \Rightarrow 0 < \phi_\epsilon^{norm}(z) < 1, z \in \mathbb{R} \text{ (Domain of Function Image)} \quad (7)$$

$$1 > \phi_\epsilon^{norm}(z) \geq p \approx 1 \iff p \approx 1 \text{ and } z \leq 0 \text{ (by } \Phi(-\frac{z}{\epsilon} + \Phi^{-1}(p)) \geq \Phi(\Phi^{-1}(p)) = p) \quad (8)$$

$$\phi_\epsilon^{norm}(z) \leq 1 - p \approx 0 \iff p \approx 1 \text{ and } z \geq \delta(\epsilon) = 2\epsilon \cdot \Phi^{-1}(p) \quad (9)$$

Proof. For $\delta(\epsilon) = 2\epsilon \cdot \Phi^{-1}(p)$

$$\phi_\epsilon^{norm}(\delta(\epsilon)) = \Phi(-\frac{\delta(\epsilon)}{\epsilon} + \Phi^{-1}(p)) = \Phi(-2 \cdot \Phi^{-1}(p) + \Phi^{-1}(p)) = \Phi(-\Phi^{-1}(p)) = 1 - \Phi(\Phi^{-1}(p)) = 1 - p$$

and from strictly monotonically decreasing we see

$$z \geq \delta(\epsilon) \Rightarrow \phi_\epsilon^{norm}(z) \leq \phi_\epsilon^{norm}(\delta(\epsilon))$$

The $\Rightarrow$-direction follows from strict monotonicity of ϕ_ϵ^{norm} and thus injectivity - i.e. $\phi_\epsilon^{norm}(z) = \phi_\epsilon^{norm}(z') \Rightarrow z = z'$. □

The limit properties are very simple to show. Since Φ is a proper continous cdf[11] and $-\frac{z}{\epsilon} + \Phi^{-1}(p) \to -\infty$ for $\epsilon \to 0$ and any fixed $z > 0$ it follows directly $\lim\limits_{\epsilon \to 0} \phi_\epsilon^{norm}(z) = 0$. Also for any fixed $z < 0$ the same properties directly establish $\lim\limits_{\epsilon \to 0} \phi_\epsilon^{norm}(z) = 1$. As a consequence the indicator can be approximated arbitrarily close when setting ϵ. Only for $z = 0$ this is cannot be established since for that case the function $\phi_\epsilon^{norm}(0) = p$ by design. Also the function $\phi_0^{norm}(z)$ has to be redefined since at $z = 0$ it is not adequately defined. Since we want it to be equal to the indicator funtion $\mathbb{1}(z \leq 0)$ we set $\phi_0^{norm}(0) \coloneqq 1$ - the limit from the left ($\lim\limits_{z \nearrow 0} \phi_0^{norm}(z)$). For the cases $z \in \mathbb{R} \setminus \{0\}$ it suffices to set $\phi_0^{norm}(z) = \lim\limits_{\epsilon \to 0} \phi_\epsilon^{norm}(z)$. Since machine precision is finite anyway these theoretical elaborations have no impact on the application that uses an adequate $\epsilon > 0$ so that e.g. the derivatives remain finite.

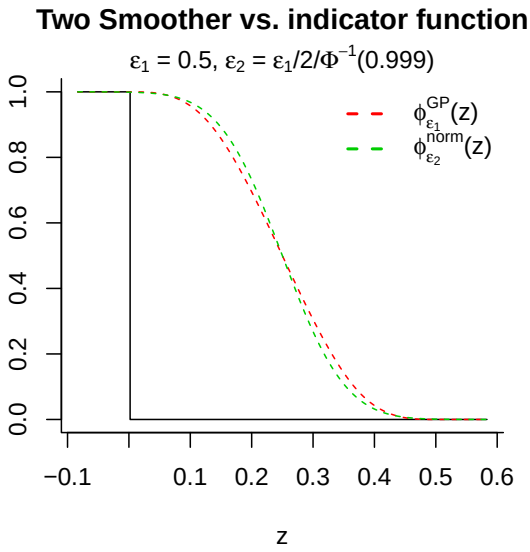
Tcd follows from (infinite) differentiability of Φ and the fact that $z = \xi^t x - \theta$ is linear with respect to x and θ (chain rule).

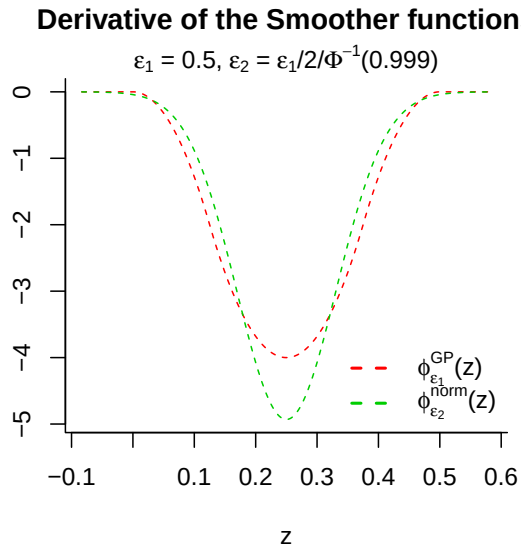
The properties shown in (8) and (9) refer to the domain of the image. Since the above defined function ϕ_ϵ^{norm} is controllable via p the deviation from the domain $[0, 1]$ is controllable as well. For a choice of e.g. $p = 0.999$ the impact of the smaller function image - i.e. $[1 - p, p]$ - is negligible. Especially when taking into account that the overall smoothing function $S_\epsilon(\theta, x)$ includes a division by the number of scenarios (e.g. 500 or 1000). In the worst case[12] - i.e. equal (or nearly equal) return values in all scenario for a fixed portfolio mix - the deviation from $\hat{F}(\theta)$ is $1 - p$. The deviation introduced by smoothing is disproportionally higher. This is why this deviation is mentioned but in the further sections not accounted for.

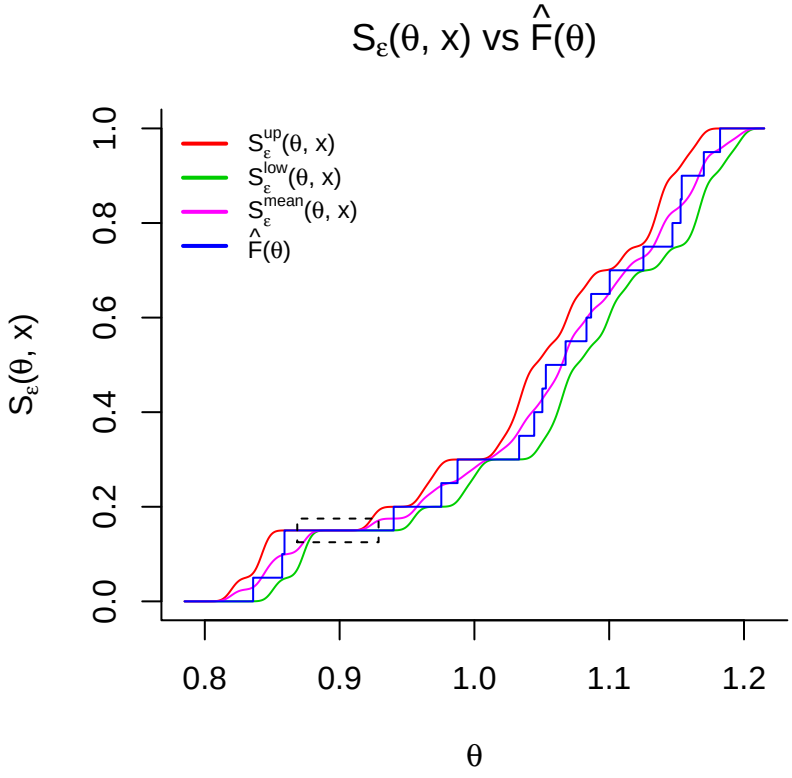
[11] Two properties of every cdf - by definition - are $\lim\limits_{z \to -\infty} F(z) = 0$ and $\lim\limits_{z \to \infty} F(z) = 1$.

[12] The worst case scenario is easy to see since the maximum deviation is attained when $\theta = \xi_j x, \forall j$ and therefore $\hat{F}(\theta) = 1$ but $S_\epsilon(\theta, x) = \frac{mp}{m} = p$. Such a constellation would only be possible if all scenarios would have a common intersection. Such a scenario is very unlikely in any reasonable application.

In the following graphs one can compare the functions ϕ_ϵ^{GP} and ϕ_ϵ^{norm} and the respective derivatives for comparable values of ϵ[13]. The full approximation $S_\epsilon(\theta, x)$ for a fixed x and using the smoother ϕ_ϵ^{norm} can be seen in the third subsequent graph.





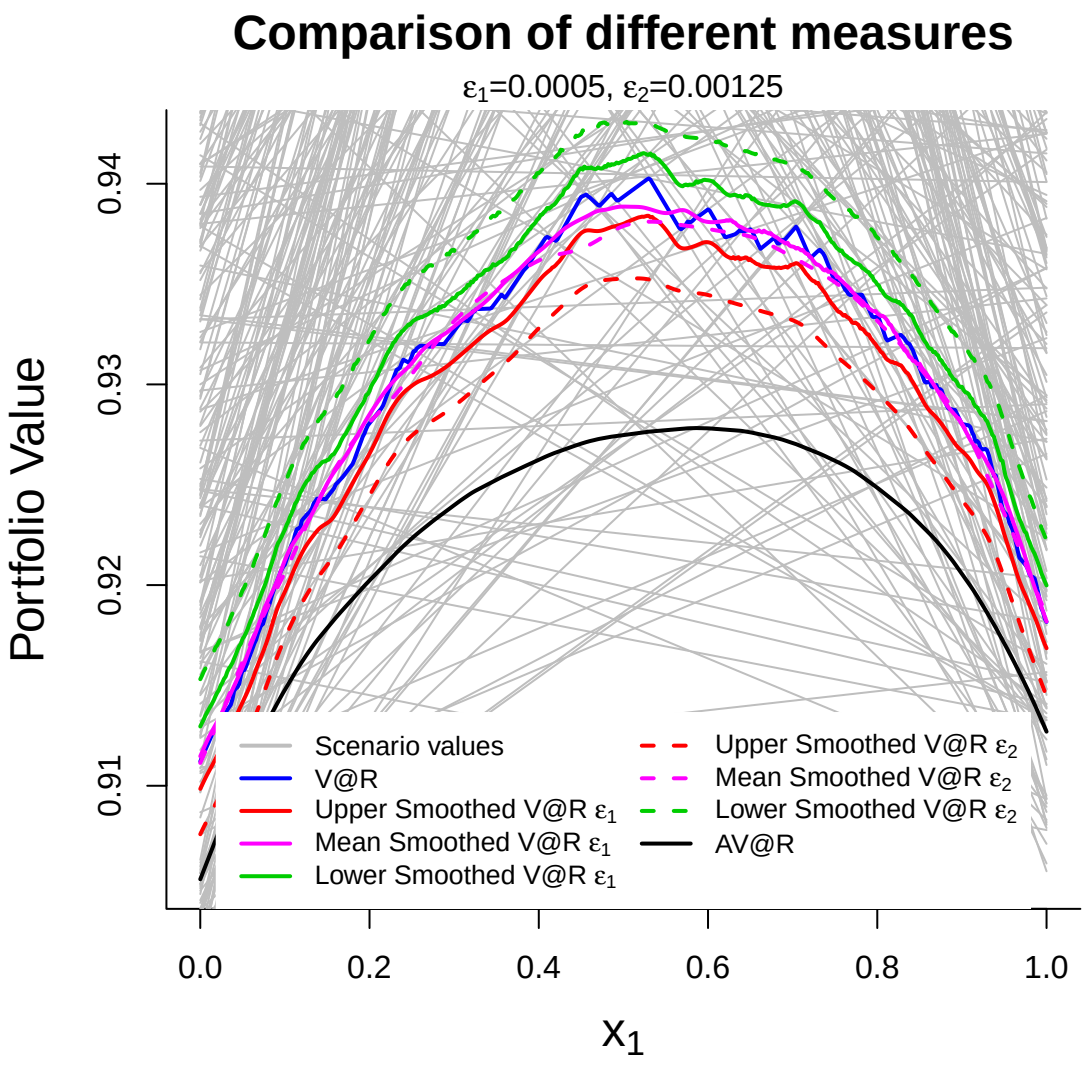


[13] $\phi_{\epsilon_1}^{GP}(z) = 0 \Leftrightarrow z = \epsilon_1$; $\phi_{\epsilon_2}^{norm} \approx 0 \Leftrightarrow z = \delta(\epsilon) = 2\epsilon_2 \cdot \Phi^{-1}(p)$ and therefore $\epsilon_1 = \delta(\epsilon) \Leftrightarrow \epsilon_2 = \epsilon_1/2/\Phi^{-1}(p)$

The green and the red lines could e.g. be used as upper and lower bounds for the V@R. It can also be observed that the approximations $S_\epsilon(\theta, x)$ are quasilinear functions with respect to θ. Since the function $S_\epsilon(\theta, x)$ varies also with x and $\xi_{\cdot,\cdot}$ this can not be exploited.

In the above graph also a problematic area is emphazized by the black dashed rectangle. Here the optimation result for θ may not be the (leftmost) minimal value. Since such arreas are more prominent for the upper and lower approximations (red and green) than for θ derived from the mean of both (violet) the final smoothed empirical cdf in the later sections will be the following: $S_\epsilon^{mean}(\theta, x) := (S_\epsilon^{up}(\theta, x) + S_\epsilon^{low}(\theta, x))/2$. One drawback is that this smoothed empirical cdf is not a clear upper or lower bound anymore but information below and above θ are used in the derivative.

In a two dimensional setting one can see the behavior of the smoothed V@R θ derived by applying (1), ϕ_ϵ^{norm} and varied x (only x_1 is given since $x_2 = 1 - x_1$).



Derivatives of $S_\epsilon(\theta, x)$

Most software capable of solving NLPs also offer algorithms to numerically approximate the derivative. In a multidimensional setting and when difficult functions are used this may be a valid option. It has to be stressed though that such numerical approximations can be - since they evaluate the functions in question multiple times - computationally expensive and the quality of the approximation is a matter of multiple parameters (e.g. machine precision, parameters of the approximation, shape of the function, type of derivative implemented[14], even software and package versions). For the function at hand the derivatives are rather simple to obtain and their use gives a tremendous performance boost.

The approximation $S(\xi_{\cdot,\cdot}, x, \theta, \epsilon)$ depends upon two variables that are varied in the optimization - i.e. x and θ. The other parameters are the data - i.e. $\xi_{j,i}$ - and the chosen smoothing parameter ϵ. Additionally $\mathbb{1} \in \mathbb{R}^m$ is the m-dimensional 1-vector. For the necessary parameters the derivatives are given by:

$$\frac{\partial S(\xi_{\cdot,\cdot}, x, \theta, \epsilon)}{\partial x_i} = \frac{1}{m}\sum_{j=1}^{m} \frac{\partial \phi_\epsilon(z_j^{(x,\theta)})}{\partial z_j^{(x,\theta)}} \cdot \frac{\partial z_j^{(x,\theta)}}{\partial x_i} = \frac{1}{m}\sum_{j=1}^{m} \frac{\partial \phi_\epsilon(z_j^{(x,\theta)})}{\partial z_j^{(x,\theta)}} \cdot \xi_{j,i} = \frac{1}{m}\xi'_{\cdot,i} D_{\phi_\epsilon}(z^{(x,\theta)}) \tag{10}$$

$$\frac{\partial S(\xi_{\cdot,\cdot}, x, \theta, \epsilon)}{\partial \theta} = \frac{1}{m}\sum_{j=1}^{m} \frac{\partial \phi_\epsilon(z_j^{(x,\theta)})}{\partial z_j^{(x,\theta)}} \cdot \frac{\partial z_j^{(x,\theta)}}{\partial \theta} = \frac{1}{m}\sum_{j=1}^{m} \frac{\partial \phi_\epsilon(z_j^{(x,\theta)})}{\partial z_j^{(x,\theta)}} \cdot (-1) = -\frac{1}{m}\mathbb{1}' D_{\phi_\epsilon}(z^{(x,\theta)}) \tag{11}$$

$D_{\phi_\epsilon}(z)$ is strictly speaking not correct since the function ϕ_ϵ previously was defined as one dimensional function $\phi_\epsilon : \mathbb{R} \to [0, 1]$. Setting $\phi_\epsilon(z) = (\phi_\epsilon(z_1), ..., \phi_\epsilon(z_m))'$ the simplification of the notation does not change any of the relevant properties but allows to write the Jacobian matrix containing all partial derivatives more compactly:

$$D_{S_\epsilon}((\theta, x)) = \frac{1}{m}\begin{pmatrix} -\mathbb{1}' D_{\phi_\epsilon}(z^{(x,\theta)}) \\ \xi'_{\cdot,\cdot} D_{\phi_\epsilon}(z^{(x,\theta)}) \end{pmatrix} = \frac{1}{m}(-\mathbb{1}, \xi_{\cdot,\cdot})' D_{\phi_\epsilon}(z^{(x,\theta)}) \tag{12}$$

$(-\mathbb{1}, \xi_{\cdot,\cdot})$ is the data matrix complemented by -1 in the first column. This gradient is very simple and only contains m function evaluations and a matrix multiplication. As a consequence the NLP can be solved extremly efficient and is very flexible in terms of the function ϕ_ϵ.

For the approximation from below via $1 - \phi(-z)$ the derivative is the same as shown before.

Proof.

$$\frac{\partial\ 1 - \phi_\epsilon(-z)}{\partial z} = -\frac{\partial \phi_\epsilon(-z)}{\partial z} \cdot (-1) = \frac{\partial \phi_\epsilon(-z)}{\partial z}$$

□

[14] e.g. Newtons difference quotient or symmetric difference quotient - the latter is used in the applied package 'nloptr'

The derivative of (the univariate) ϕ_ϵ^{norm} with respect to z is also rather simple. Denoting the standard normal pdf as $N(x)$ the derivative is given by:

$$\frac{\partial \phi_\epsilon^{norm}(z)}{\partial z} = -\frac{1}{\epsilon} N(-\frac{z}{\epsilon} + \Phi(p)) \tag{13}$$

For ϕ_ϵ^{GP} the derivative is:

$$\frac{\partial \phi_\epsilon^{GP}(z)}{\partial z} = \begin{cases} -\frac{16}{\epsilon^3} z^2 & z \in (0, \frac{\epsilon}{4}] \\ \frac{2}{\epsilon} - \frac{16}{\epsilon^2} z + \frac{16}{\epsilon^3} z^2 & z \in (\frac{\epsilon}{4}, \frac{3\epsilon}{4}] \\ -\frac{16}{\epsilon} + \frac{32}{\epsilon^2} z - \frac{16}{\epsilon^3} z^2 & z \in (\frac{3\epsilon}{4}, \epsilon] \\ 0 & else \end{cases} \tag{14}$$

At this point all necessary prerequisites are established and the NLP can be tested via a simulated data set.

Application to Data

Overview

In this section the above developed NLP with the smoothed V@R will be illustrated with several examples.

At first the feasible solutions will be plotted in a setting of three variables. In this setting also a first glance on how a derivative based algorithm approaches the final solution is shown graphically. Concluding this first part the found solutions are also plotted. Since this section is for illustrative purposes only, no detailed description of the setting is given (it is similar to the setting of the subsequent practical demonstration).

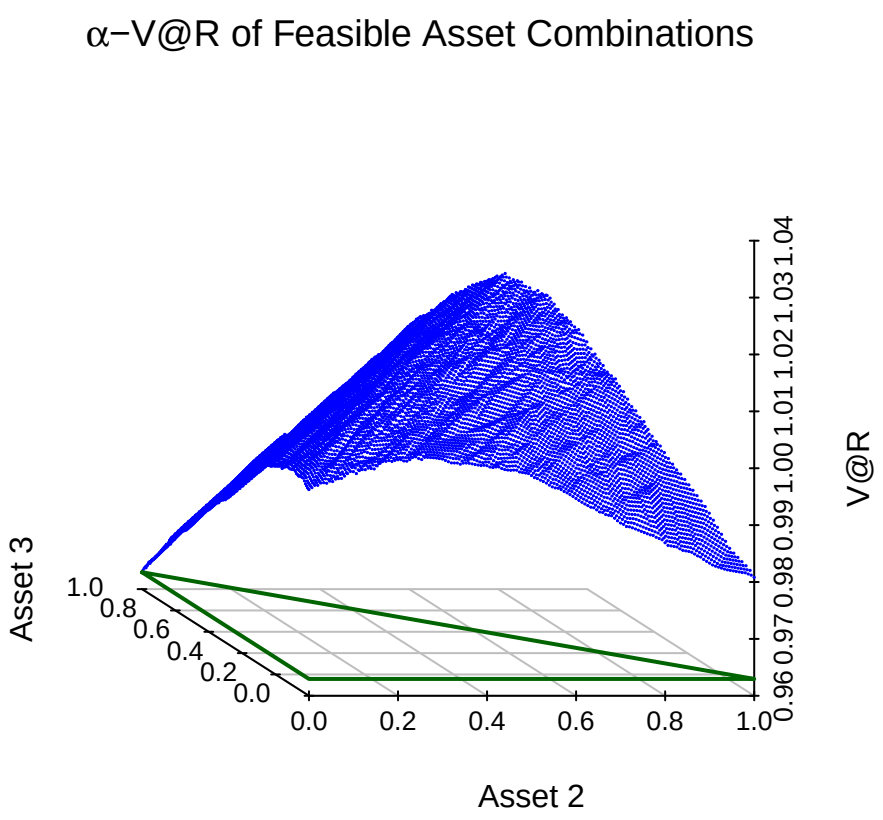
In the (first) practical demonstration a rather intuitive and comprehensive framework has been chosen. In other words, a setting where all conditions can be listed and controlled seems appropriate. Therefore the first application uses simulated data from several triangular distributions and - to make things a little bit more realistic - the random variables may be (linearly) correlated with a random correlation matrix. In this setting some illustrative graphs are given and several algorithms and risk measures will be compared to each other.

In a first step the setting and the smoothed V@R will be illustrated. In the next step several algorithms will be compared in terms of performance and adequacy. Finally - for the random data set - the solutions based on different risk measures will be compared.

To demonstrate that the algorithm also works in a real world setting in the final section an application to such a data set is given.

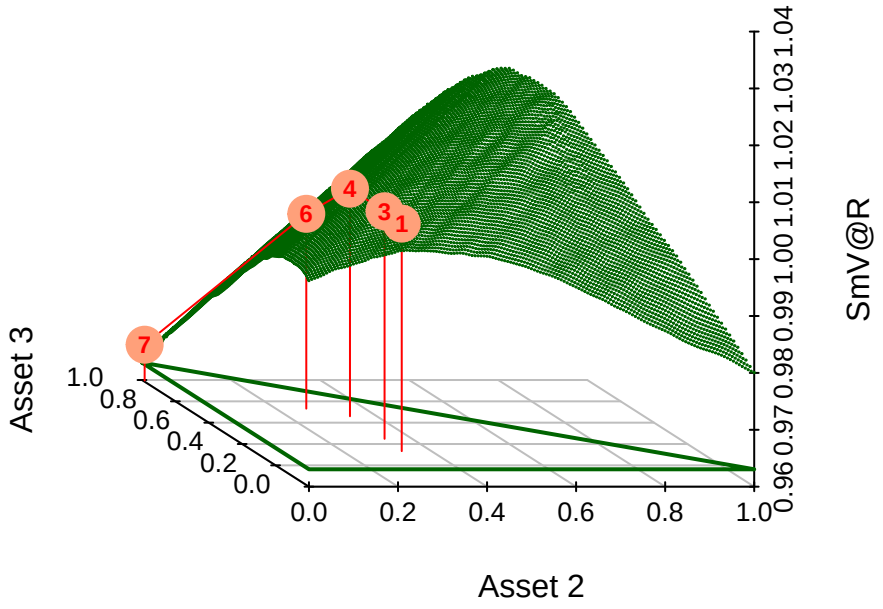
Illustration of Setting and Algorithm

In a three dimensional setting - i.e. three possible assets - the space of feasible solutions can be seen in the subsequent graph. The blue surface is the real $V@R_\alpha$ of the feasible asset combinations $\mathbb{1}'x = 1$ (bounded by the green frame). Since $x_1 = 1 - x_2 - x_3$ and the "most interesting" assets are asset 2 and 3 the x_1 axis was excluded to "fit" into a 3D-graph. It can be seen that the surface of the $V@R_\alpha$ is pretty wrinkled or "rough".



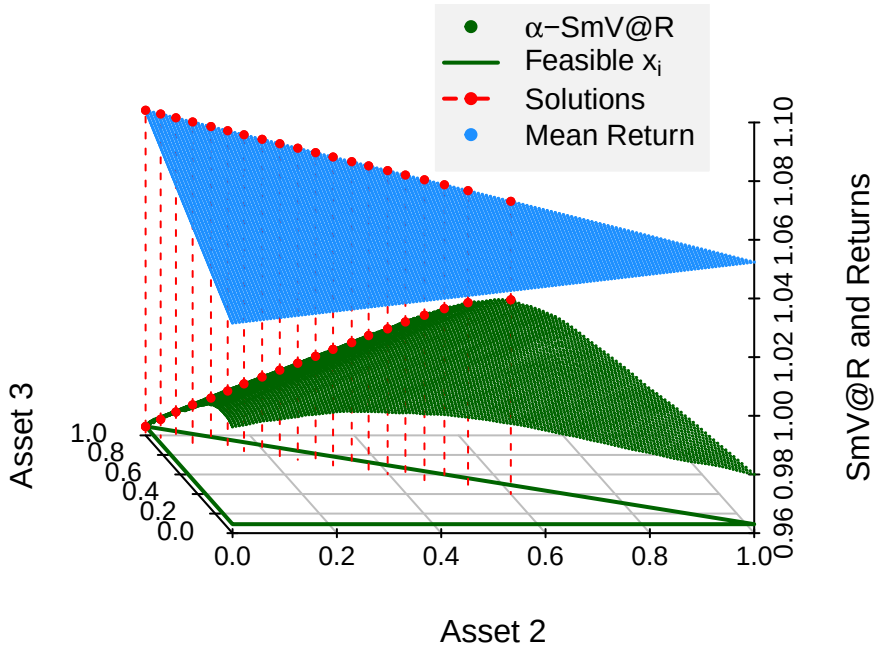
In the next graph one can see the smoothed V@R (called smV@R and plotted in green) and the steps of a derivative based NLP algorithm towards the final solution. Starting with an equally distributed portfolio (step 1) the algorithm quickly finds the proper descent direction and after only a few steps the arrea with the most promising solution is found (step 7). The rest of the search is a local one and after 30 steps a local minimum within the feasible space is found.

Feasible Space and Optimizer Steps

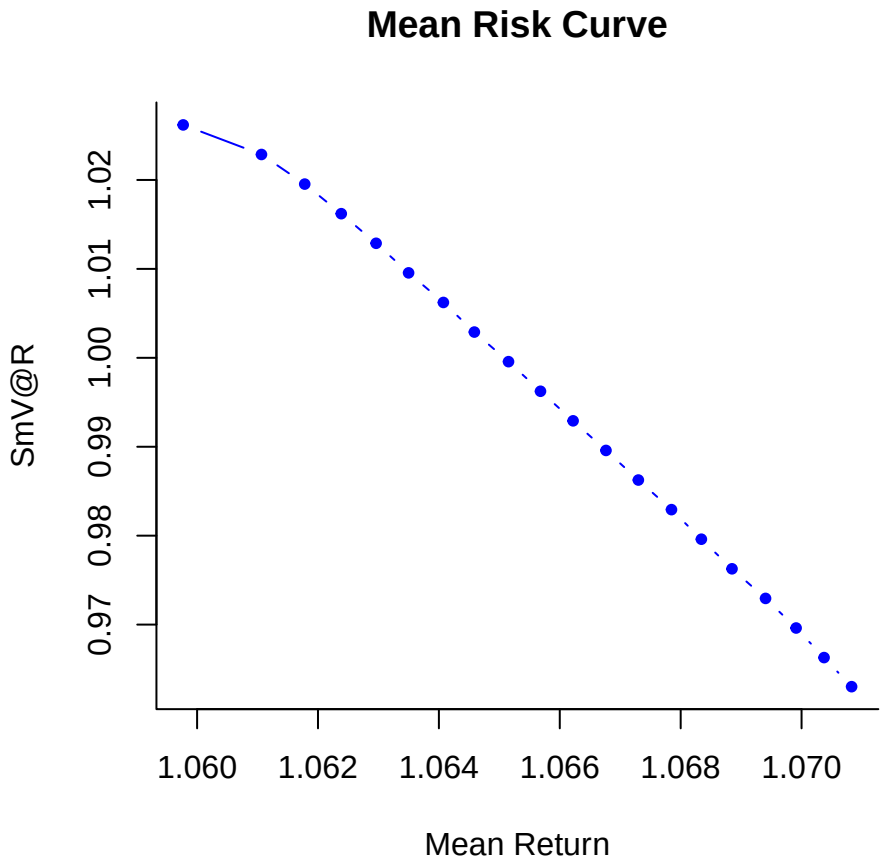


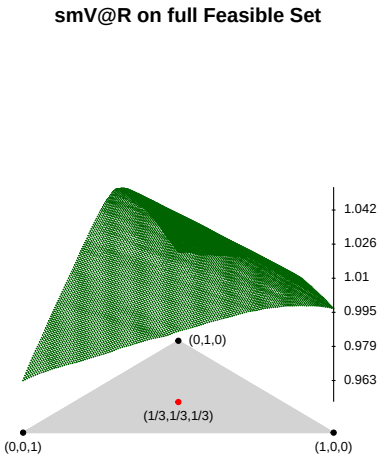
Step Sequence	Theta	Asset 1	Asset 2	Asset 3
1	1	0.3333	0.3333	0.3333
3	1	0.2134	0.338	0.4486
4	1	0	0.3397	0.6603
6	0.9942	0	0.2683	0.7317
7	0.9663	0	0.00432	0.9957
30	0.9663	0	0.02473	0.9753

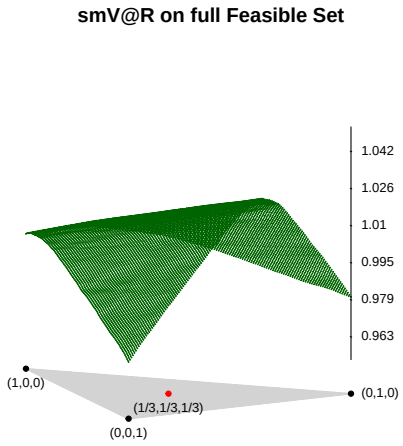
On the next page one can see the $smV@R_\alpha$ as well as the sum constraint (green) and the objective function (blue). The solutions with a varied $smV@R_\alpha$ bound are included as red points with a red dotted line for better identification of the corresponding function values (objective as well as constraints). It is worthwhile mentioning that the given information is usually shown in a mean-risk curve (see graph below the table). Since the third dimension of Asset 1 can not be seen in the first plots two additional plots are given showing the $smV@R_\alpha$ on the full feasible set and from two different angles.



Asset 1	Asset 2	Asset 3	Expected Return	Smoothed V@R	V@R
0	0	1	1.071	0.963	0.963
0	0.02473	0.9753	1.07	0.9663	0.9664
0	0.04983	0.9502	1.07	0.9696	0.9692
0	0.07726	0.9227	1.069	0.9729	0.973
0	0.1074	0.8926	1.069	0.9763	0.9767
0	0.1349	0.8651	1.068	0.9796	0.9798
0	0.1617	0.8383	1.068	0.9829	0.9824
0	0.1915	0.8085	1.067	0.9863	0.9862
0	0.2205	0.7795	1.067	0.9896	0.9901
0	0.25	0.75	1.066	0.9929	0.9925
0	0.2793	0.7207	1.066	0.9962	0.9959
0	0.308	0.692	1.065	0.9996	0.9992
0	0.3387	0.6613	1.065	1.003	1.003
0	0.3665	0.6335	1.064	1.006	1.006
0	0.3977	0.6023	1.063	1.01	1.009
0	0.4269	0.5731	1.063	1.013	1.013
0	0.4582	0.5419	1.062	1.016	1.017
0	0.4909	0.5091	1.062	1.02	1.02
0	0.5298	0.4702	1.061	1.023	1.023
0	0.6001	0.3999	1.06	1.026	1.026


Mean Risk Curve
SmV@R
0.97
0.98
0.99
1.00
1.01
1.02
1.060
1.062
1.064
1.066
1.068
1.070
Mean Return


smV@R on full Feasible Set
1.042
1.026
1.01
0.995
0.979
0.963
(0,1,0)
(1/3,1/3,1/3)
(0,0,1)
(1,0,0)


smV@R on full Feasible Set
1.042
1.026
1.01
0.995
0.979
0.963
(1,0,0)
(1/3,1/3,1/3)
(0,1,0)
(0,0,1)

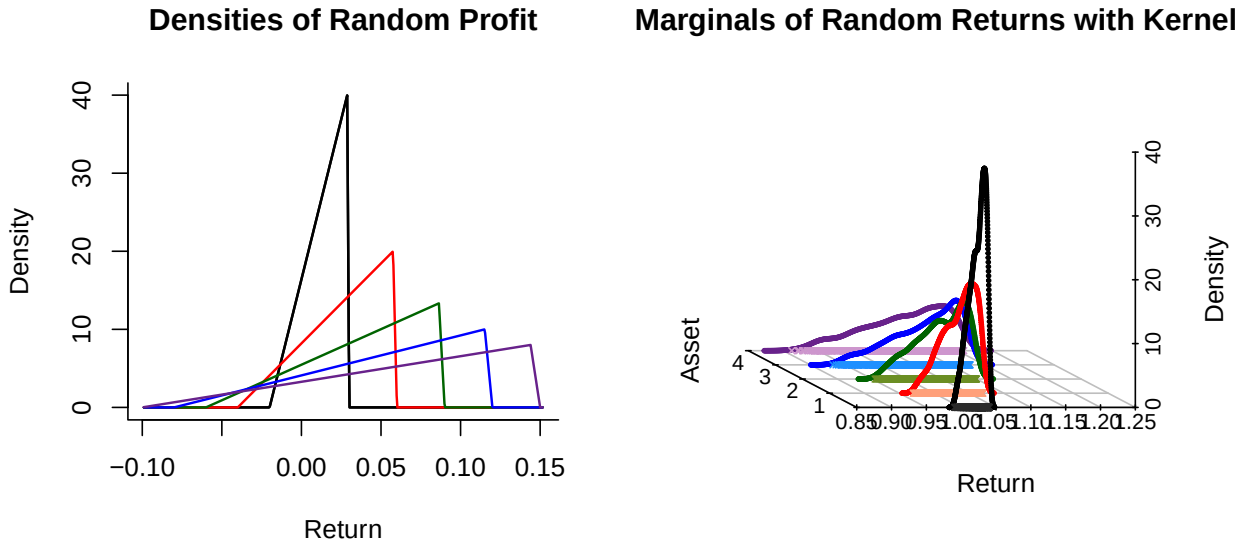
Simulated Data

Description of Data

The simulated data set will be drawn from the triangular distribution with varied parameters. The reasoning behind is that the triangular distribution can be parametrized very intuitively and the created data looks similar to a log-normal distribution (pretty often used for log returns) while beeing bounded. To be more specific the returns of the asset (can) have a skewed distribution and the mean as well as the variance are given straight forward.

Since the optimization task could be simplified a lot in a setting where the returns are independent a linear dependence structure is added to avoid that neglecting such a dependence could perturb the drawn conclusions significantly. Even for a triangular distribution this can be a non-trivial task but via the NORATA algorithm ("Normal-to-Anything")[15] almost arbitrary correlation structures can be simulated (taking into account the prerequisites of a proper correlation matrix, e.g. the matrix beeing non-negative definite and symmetric). It should be mentioned here that introducing a dependence structure for general random variables is a rather simple task but maintaining the marginal distributions when doing that can be hard to achieve. In the below section one can see the data parameters, the (closed form) densities, kernel estimates and correlcation structure.

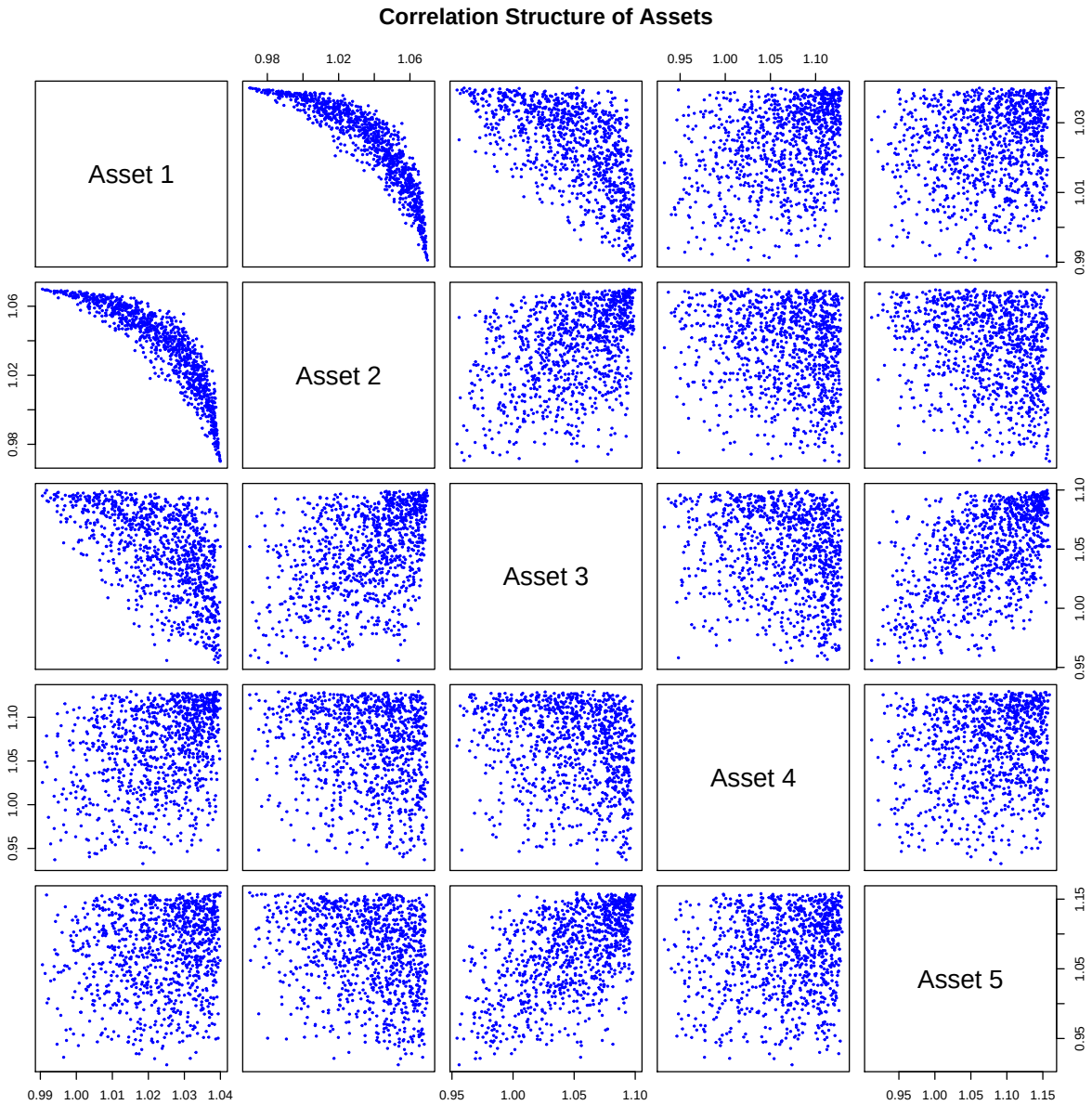
	a	b	c	E(X)	Sample Mean	SD(X)	Sample Std. Dev.
Asset 1	0.99	1.04	1.039	1.023	1.023	0.01165	0.01173
Asset 2	0.97	1.07	1.068	1.036	1.035	0.02329	0.02373
Asset 3	0.95	1.1	1.096	1.049	1.047	0.03494	0.03551
Asset 4	0.93	1.13	1.125	1.062	1.065	0.04659	0.04708
Asset 5	0.91	1.16	1.154	1.075	1.074	0.05823	0.0582



[15] see R package documentation for "NORTARA" and the article Niavarani/Smith (2012)

Table 7: Correlation estimates of Assets

	Asset 1	Asset 2	Asset 3	Asset 4	Asset 5
Asset 1	1	-0.875	-0.592	0.321	0.158
Asset 2	-0.875	1	0.41	-0.133	-0.224
Asset 3	-0.592	0.41	1	-0.191	0.553
Asset 4	0.321	-0.133	-0.191	1	0.167
Asset 5	0.158	-0.224	0.553	0.167	1



Optimization Setup

The data shown above is used for the optimization task described in the first sections of this thesis. Several questions arise now when it comes to the optimization algorithms to use to solve the NLP:

$$\begin{aligned} \underset{(\theta,x)\in\mathbb{R}\times\mathbb{R}^{n+}}{min} \quad & -\bar{\xi}'x \\ \text{s.t. } & S(\xi_{\cdot,\cdot},x,\theta,\epsilon) \coloneqq \frac{1}{m}\sum_{j=1}^{m}\phi_\epsilon(\xi'_{j,\cdot}x-\theta) \leq 1-\alpha \\ & \theta_{min} \leq \theta \leq \theta_{max} \\ & \mathbb{1}'x \leq 1 \end{aligned}$$

The first one is obviously the question if the property of tcd is necessary at all or if non-derivative based algorithms give similar results with higher freedom in terms of specification. So one point of the comparison in the next section is also to show that non-derivative-based algorithms (like COBYLA and BOBYQA) don't perform as good as derivative-based algorithms.

For the following conclusions it has to be mentioned that running an optimization with general non-linear constraints can be a challenge on its own and small changes can have big effects. Therefore the conclusions drawn here are only valid for the setting at hand and should not be interpreted as beeing fully generalizable.

There are plenty of optimization algorithms available that differ in method, needed information (e.g. derivative) and specification (e.g. problem setup, parameters). Most importantly it should be mentioned that all herein inspected algorithms are based on the Lagragian optimization approach. To be more specific they use an Augmented Lagrangian problem formulation and even within this subset of NLP algorithms the quality of the obtained solutions is subject to a multitude of specification options (e.g. problem implementation, software or packages, sub-algorithm(s), termination criteria, penalty parameters for constraint violation, techniques to avoid or correct violations of constraints within one optimization run).

In the next table several algorithms are presented that were investigated and experimentally used but were finally **NOT** included. Also the reason for exclusion is given.

Algorithm	**Reason of Exclusion**
random search	always applicable and thus often used as benchmark model but specifically bad in problems where the search space is "big"; since the presented method is built for a more guided search and outperforms random search when it comes to found solutions, violated constraints and run-time it has been excluded
evolutionary/genetic algorithms	non-standard method better suited for complex setups where the functional specification is challenging; quality of solution and run-time is subject to the problem formulation which brings high effort; standard implementations of those algorithms were not performing well, therefore it is not selected here
Nelder-Mead	applied implementations had high run-time and frequently violated given constraints; high dependence on starting points and polytope specification

The R software provides several packages that can be used for optimization. Most of them are specialized on a certain subject or technique (e.g. linear or quadratic optimization or NLP with box-constraints). The applied package to derive the below results was the R-package "nloptr". The NLOPTR-framework offers a broad bandwidth of applicable algorithms, appears to be implemented well and offers flexible parameter specification. The algorithms compared later in this section are listed in the next table (see NLopt-Wiki Algorithms on http://ab-initio.mit.edu/wiki/index.php/NLopt_Algorithms for more details; the quoted short descriptions are taken directly from this link).

Algorithm	**Short Description**	**Derivative Based**
Sequential Least-Squares Quadratic Programming (SLSQP)	A variant of Sequential Quadratic Programming (SQP) that is frequently used for NLPs. The quoted page includes a warning for big problems (more than "a few thousand parameters") since it uses BFGS and not low-storage BFGS.	Yes
Preconditioned Inexact Truncated Newton Algorithm (TNEWTON)	This variant of the Newton method allows for preconditioning "by low-storage BFGS algorithm with steepest-descent restarting". It can also be used with preconditioning, restarting or both.	Yes

Algorithm	**Short Description**	**Derivative Based**
Method of Moving Asymptotes (MMA)	"[G]lobally-convergent method-of-moving-asymptotes (MMA) algorithm for gradient-based local optimization, including nonlinear inequality constraints (but not equality constraints)[.] ... At each point **x**, MMA forms a local approximation using the gradient of f and the constraint functions, plus a quadratic 'penalty' term to make the approximations 'conservative' (upper bounds for the exact functions). [...] [I]t includes nonlinear terms consisting of a poles at some distance from x (outside of the current trust region) [...]. The main point is that the approximation is both convex and separable, making it trivial to solve the approximate optimization by a dual method."	Yes
Low-storage BFGS (LBFGS)	This algorithm is a quasi-Newton method and is one of the most common NLP algorithms (usually for uncostraint NLPs). The full name is Broyden-Fletcher-Goldfarb-Shanno algorithm and uses a special technique to approximate the Hessian matrix where the derivative information of **m** subsequent steps is memorized for a better approximation of the Hessian. The "low-storage" version allows a more efficient memory usage when optimizing problems with more than 1000 variables.	Yes
Constrained Optimization by Linear Approximations (COBYLA)	This algorithm "constructs successive linear approximations of the objective function and constraints via a simplex of n+1 points (in n dimensions), and optimizes these approximations in a trust region at each step."	No
Derivative free Quadratic Approximation (BOBYQA)	"BOBYQA performs derivative-free bound-constrained optimization using an iteratively constructed quadratic approximation for the objective function. [...] Because BOBYQA constructs a quadratic approximation of the objective, it may perform poorly for objective functions that are not twice-differentiable."	No

The algorithms above are applied as the inner optimization algorithms of the Augmented Lagrangian algorithm - allowing arbitrary non-linear constraints. Some differences were experienced between explicit and implicit accounting for equality constraints - i.e. “AUGLAG_EQ” vs. “AUGLAG” - despite the fact that no equality constraints were implemented. The reason for this remains unclear but is the basis why for “SLSQP”, “MMA” and “COBYLA” the method “AUGLAG_EQ” is used while the others were combined with “AUGLAG”. This appeared to make the runs more stable and increase performance.[16]

The above algorithms will be compared in terms of:

- solution (expected return and V@R)
- susceptibility to errors
- constraints violation
- dependence on starting points
- run-time

The best algorithm with respect to the above criteria will be compared to the AV@R and Markowitz approach - at first with the simulated data set and in the next section with real return data. The above criteria should be measured as objective as possible and will be analyzed in the following way:

Solution: The quality of solution cannot be directly measured objectively but by a comparison of the Mean-Risk-Curves a benchmarking of the algorithms is possible. Therefore the subject here is to show the results and get insights into some weaknesses of the algorithms under investigation.

Susceptibility to Errors: Experience shows that algorithms (or their implementations) vary significantly concerning the susceptibility to errors during one optimization run. This results in early termination and erroneous outcomes. For the task at hand it is necessary to have a reliable and stable algorithm that gives reasonable results. In the table of comparison in the "Results" section this is measured as proportion of runs with some sort of failed status at termination to all runs.[17]

Constraint Violations: One general problem in NLP is the possible violation of constraints in every iteration and also at the final solution. Since optimal solutions are frequently found at the margin of the feasible space or algorithms may leave the feasible space without beeing able to re-enter it (e.g. any termination criterion that aims at changes of the solution or the function arguments x can lead to a termination outside the feasible space) it should be checked separately if the result is feasible. A constraint violation flag is assigned if either $1 - \mathbb{1}'x > 10^{-5}$, $\theta_{min} - \theta > 10^{-7}$ or $S(\xi_{.,.}, x, \theta, \epsilon) - \alpha > 10^{-7}$. This is sufficient since all values should be negative or 0 anyway. The runs were counted excluding failed runs so the presented proportion is a conditional proportion.

[16] Since the converse was observed as well in other settings this decision is not necessarily true in all settings.

[17] In the used R-package "nloptr" a positive status code means positive termination status. The status "-4" on the other hand flags a failed status that is triggered when numerical problems - i.e. machine precision - does not suffice for any further progress. This status usually also gives reasonable results and is thus counted as successful run.

Dependence on Starting Points: How much the final solutions vary with the starting point x_0 is a good indication for how well suited the algorithm is for finding a good "global" solution. To measure this the mean range of the final solutions with different starting points is used (mean(max - min))[18]. Failed runs again distort the measure and are excluded.[19]

Run-Time: The run-time is measured in seconds and is given as mean over all runs of an algorithm - not accounting for e.g. longer run-times in runs with more challenging constraints or possibly shorter run-times for early or erroneous termination.

The function $\phi_\epsilon^{norm}(z)$ was used as smoother with $\epsilon = 0.0005$. Every algorithm was run with five different starting points x_0 and 50 varying constraints for θ_{min} - i.e. the smV@R value. Two starting points were the solutions of the Markowitz and AV@R optimization respectively, one was the centroid (each $x_0^{(i)} = \frac{1}{n.securities}$) while the remaining two where random points drawn from an equally weighted dirichlet distribution (this is equivalent to drawing a random point fulfilling the constraint $\mathbb{1}'x = 1$ from an uniform distribution).

The θ_0 - the first component of x_0 - has been calculated before each optimization run to obtain a feasible starting point. This is not strictly necessary but simplifies the run. For reasonable values of the maximum θ_{min} again the V@R values from the Markowitz and AV@R optimization have been used. This is not necessarily the best choice.[20] After a first run with $\theta_{min} = 0$ - in this setting this is equivalent to an unconstraint run - the result was used to derive 48 equidistant θ_{min} inside the mentioned range. Therefore for every inner algorithm 250 runs have been performed in total.

Although the same experiment has been performed using $\phi_\epsilon^{GP}(z)$ as smoothing function the outcomes did not differ significantly and the algorithm performing best is the same.

Results of Algorithm Comparison

In the below plots one can see the results for each algorithm. For the truncated Newton method - "TNEWT" - the preconditioning and restart option have been abbreviated by "P" and "R" respectively.

At first glance one can see that the algorithms not using derivatives (i.e. "COBYLA" and "BOBYQA") are rather unstable and frequently violate the constraints in the final result (results with red rings).

For the truncated Newton algorithms several runs failed. This may be due to the problem specification or algorithm internal issues.

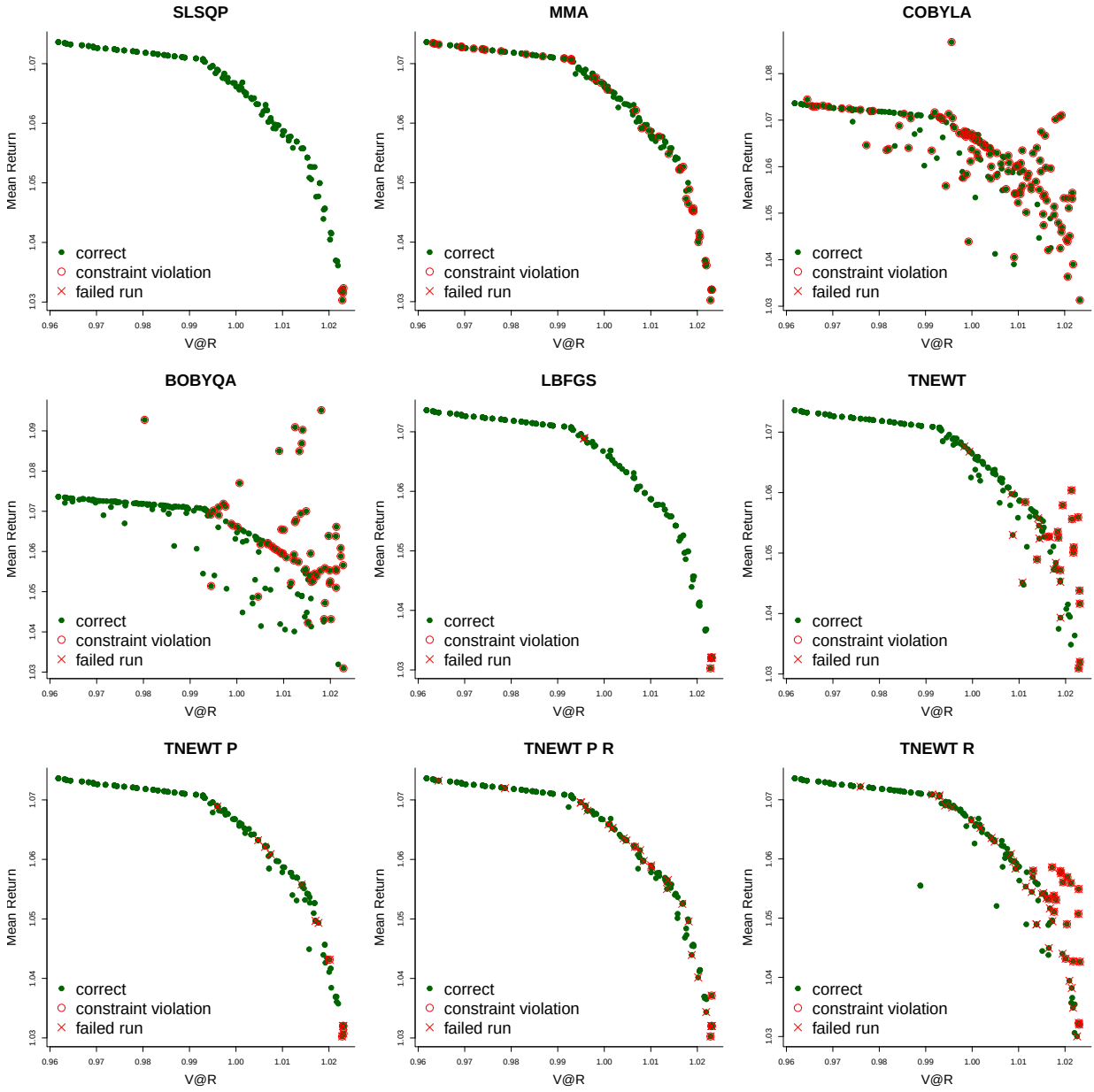
"SLSQP", "MMA" and "LBFGS" give in general good and stable optimal points but - especially "MMA" - frequently violates constraints at the final result. One reason could be that the condition for constraints violation is a little bit to strict (10^{-6} instead of 10^{-7} could suffice).

[18] The max/min is taken from different starting points and equal constraint specification and the mean is then taken over all varied constraint specifications.

[19] The exclusion also can distort the measure if e.g. only one run with equal constraint specification remains and all other are excluded. This will on the other hand be reflected in a bad performance measure in the "Susceptibility to Errors" section.

[20] The maximum θ_{min} - i.e. the highest smV@R constraint value of the 50 varying θ_{min} values - can differ from the maximum V@R and therefore the last run - with said constraint - gave no feasible solution for almost every algorithm. This intricacy was not solved since its impact is very limited.

The consistent constraints violation in the last run is strong indication that the specification of the θ_{min} in the last setting may be to strict and that the problem is actually infeasible. Depending on the algorithm this gives either a failed run or an infeasible solution.



The table below shows the ranks relative to the other algorithms and the corresponding measure in brackets - i.e. "*[rank]* (*[measure]*)". The conclusions drawn from the graphs are almost directly reflected in the mean rank and the algorithms "SLSQP" as well as "LBFGS" outperform the other algorithms in the setting at hand with respect to the inspected measures. Since "SLSQP" runs faster and more stable this algorithm will be used in the next comparison to Markowitz and the AV@R.

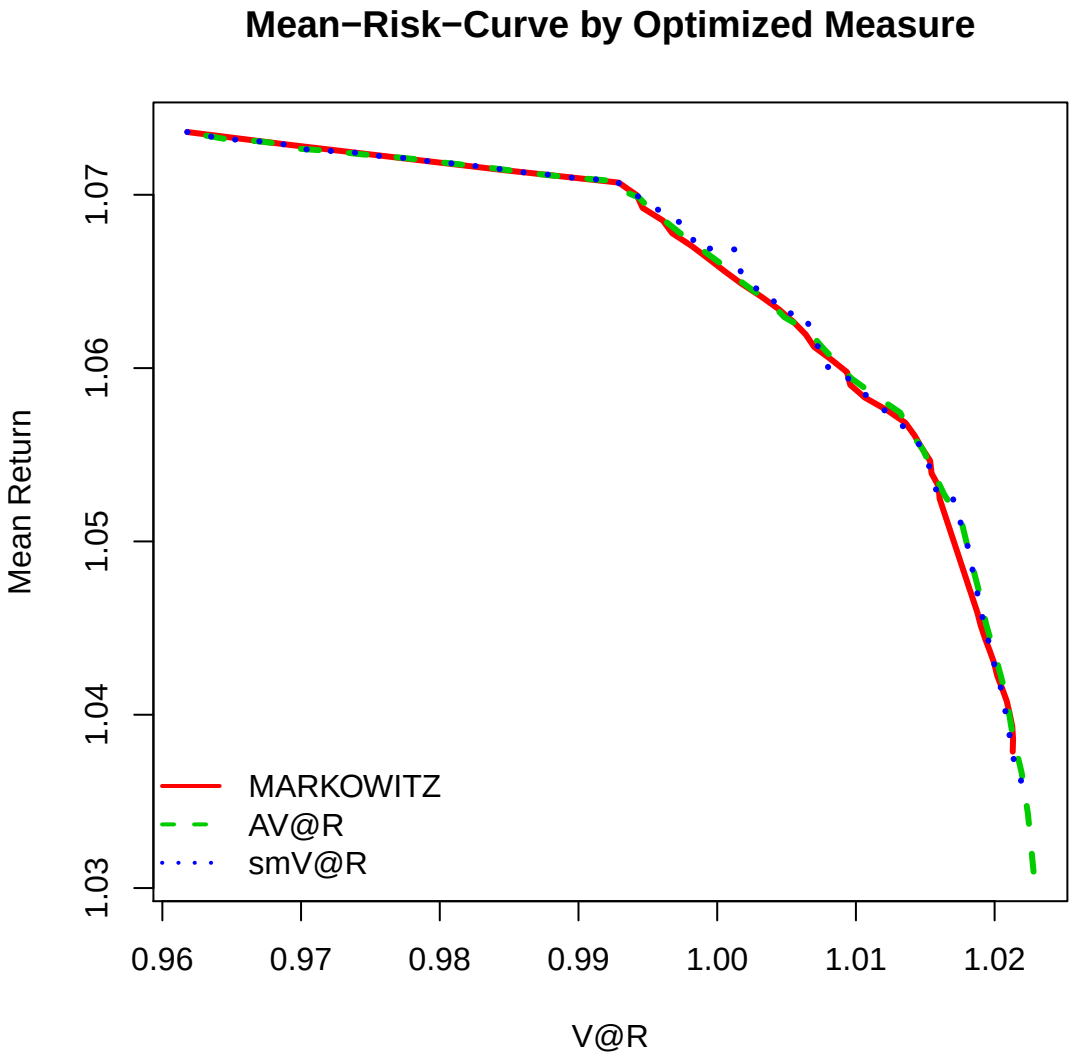
	Pct. of Runs with Errors	Constraint Violations	Dependence on x0 (Return)	Dependence on x0 (Theta)	Mean Duration	Mean Rank
LBFGS	5 (0.032)	1 (0.000)	1 (0.0002)	2 (0.0021)	3 (0.33)	2.4
SLSQP	1 (0.000)	6 (0.020)	2 (0.0004)	7 (0.0033)	1 (0.06)	3.4
TNEWTP	6 (0.052)	1 (0.000)	5 (0.0009)	3 (0.0022)	4 (0.48)	3.8
TNEWTP R	7 (0.116)	1 (0.000)	3 (0.0004)	4 (0.0022)	5 (0.54)	4
MMA	1 (0.000)	7 (0.192)	4 (0.0005)	1 (0.0003)	8 (0.93)	4.2
TNEWT	7 (0.116)	1 (0.000)	6 (0.0021)	5 (0.0023)	6 (0.69)	5
COBYLA	1 (0.000)	9 (0.472)	8 (0.0112)	9 (0.0249)	2 (0.06)	5.8
TNEWTR	9 (0.192)	1 (0.000)	7 (0.0024)	6 (0.0024)	7 (0.69)	6
BOBYQA	1 (0.000)	8 (0.260)	9 (0.0149)	8 (0.0201)	9 (5.24)	7

Results of Risk Measure Comparison

In the next plot one can see the Mean-Risk-Curves derived using different risk measures. The dotted blue line is the smV@R using "SLSQP" where points that where not at the Pareto optimal bound were deleted.[21] One can see that the curves are almost indistinguishable. So the first benchmark test is passed. In the comparison using real data the discrepancy is much more significant. At this point several conclusions can be drawn when using the other risk measures as benchmarks:

- The results obtained using the smV@R are at least as good as those from the AV@R and Markowitz approach.
- The smV@R could be optimized in terms of parameters (e.g. ϵ, x_0) used while the other two approaches give a fixed result.

[21] Inspecting the Mean-Risk-Curves at the algorithm comparison for "SLSQP" one can see that only points irrelevant for the interpretation were excluded when doing that.


Mean−Risk−Curve by Optimized Measure
Mean Return
1.03
1.04
1.05
1.06
1.07
0.96
0.97
0.98
0.99
1.00
1.01
1.02
V@R
MARKOWITZ
AV@R
smV@R

Real Data

Description of Data

In this section the above developed techniques are demonstrated for real data. For that purpose the returns on trading stocks listed in the DOW30, DAX, ATX (in total 77 possible stocks) were used.

The data source was finance.yahoo.com and the prices are already adjusted for splits and dividends (adjusted closing prices). The base data ranged from January 1^{st} 2000 to April 22^{nd} 2017 but since some stocks were not traded on a stock exchange before May 31^{st} 2006 (date of stock market launch of Oesterreichische Post AG) the data had to be truncated. Since the companies VISA and BUWOG were launched several years later and to avoid losing more data their stocks were excluded. In the case of missing data due to holidays or other reasons in one of the listed stocks the whole line has been dropped. The final stock price base data set contained 2666 lines of 77 stocks.

The data set for the optimization was built using returns - calculated by $\xi = \frac{price_{t+\delta}}{price_t}$ - from a buy-and-hold strategy for 20 business days (approximately one month).[22] Dropped lines were not considered in the assessment of the trading interval[23]. From this data set $n = 1000$ scenarios were drawn and used in the optimization.

The "SLSQP" algorithm with $\epsilon = 0.0005$ was used to calculate 20 solutions - each with different starting point - for 50 equidistant θ_{min} constraint values. As in the previous setting with random data, two starting points were taken from the solutions of the Markowitz and AV@R optimization, one was the centroid (each $x_0^{(i)} = \frac{1}{n.securities}$) while the remaining 17 where random points drawn from an equally weighted dirichlet distribution. This higher number of x_0 should account for the higher dimension of the search space.

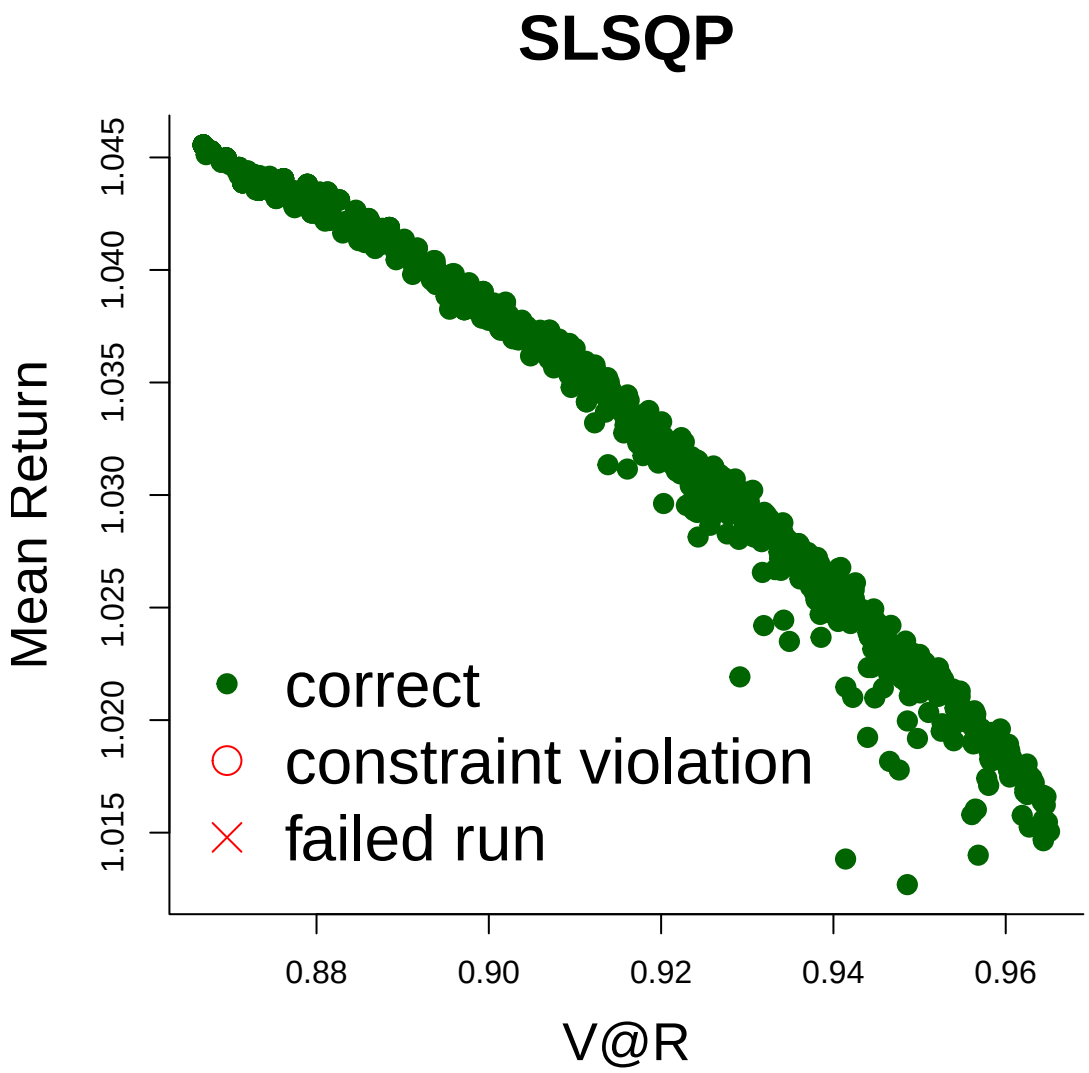
Variable	**Value**
Number of possible Investments	77
Days of Investment (δ)	20
Number of θ_{min} values	50
Number of Scenarios	1000
Number of Iterations	20
Parameter ϵ	0.0005

Results

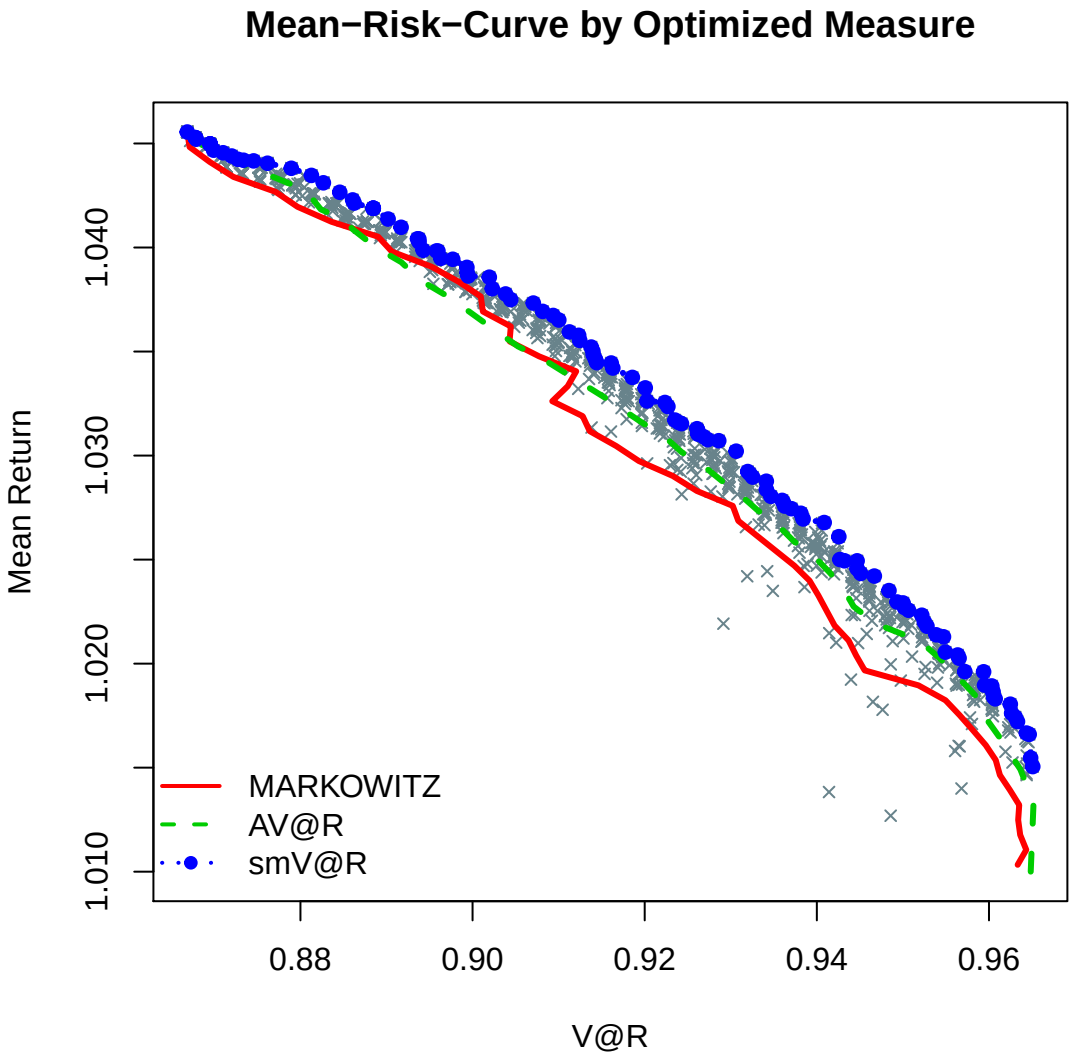
In the next graph the results of all runs have been plotted in the form of a risk curve. It can be observed that the variation - i.e. dependence on the starting points x_0 - has increased. This can mainly be attributed to the higher dimension of the optimization problem.

[22] It should be noted that the closing price not necessarily is the price one is able to sell. The deviation should be negligible for the purpose of this work.

[23] This means that the return is sometimes calculated on a minimally longer time interval than 20.



Nevertheless based on the subsequent plot it can also be concluded that the smV@R usually yields better results when the V@R is the risk measure to optimize. This can be seen by observing that most results of smV@R for different x_0 (blue crosses) are above the green line (AV@R results) and the red line (Markowitz approach). Additionally, if one takes the upper Pareto optimal bound (dark blue dots) the results obtained from the smV@R clearly yield more advantagous portfolio compositions than both of the benchmarking measures (and corresponding methods).



	NLOPT_LD_SLSQP
Susceptibility to Errors	0
Constraint Violations	0.028
Dependency on x0 (MEAN RETURN)	0.00342
Dependency on x0 (THETA)	0.00289
Sum of Run-Time (mins)	12.9
Mean of Run-Time (secs)	0.775

Discussion

In the first plot of the previous section one can see the results for the smV@R alone while in the second plot the blue crosses are all the results of the smV@R optimization runs compared to the other approaches. The blue dotted line is the pareto optimal bound derived through the smV@R with the big dots beeing the results shaping the bound.

In those above plots it becomes quite clear that the smV@R generally gives better results than the Markowitz and AV@R optimization approach. Although some variation based on the starting point x_0 can be observed for the smV@R (see first and second plot of the last section) the finally obtained upper bound is significantly better when it comes to the V@R as optimization criterion.

Also the run time (see table above) with a total of 12.9 minutes is reasonable if one considers the fact that 1000 Scenarios were used for 20 re-runs and 50 different values of θ_{min}. A mean run time of about 0.78 seconds for each optimization run is also very satisfactory - especially when considering that potentially several hundred steps/evaluations of the above functions could have been performed in each optimization.

The obvious and already mentioned downside of the presented smV@R is that one potentially has to perform multiple iterations with varying x_0. This is a consequence of the non-convexity of the smV@R and the associated potential of a large number of local optima. Since the used method concentrated on a good (i.e. close) and smooth (i.e. tcd) approximation of the V@R this intricacy was not tackled. Thus the associated method can be described as guided random search.

Closing Remarks

During the development of the above smV@R procedure a multitude of decisions had to be taken and gaps had to be closed. Many of them were only mentioned once in the above work but gave interesting insights into the problems and challenges when using the V@R as an optimization criterion. It became more obvious why the AV@R is more prominent when it comes to portfolio decisions and risk measure.

Nevertheless it should be stressed here that aside of the regulatory frameworks for the finance industry the V@R - or more the quantile - optimization in the above sense can be of high value in other fields as well. It could be used e.g. in the chemicals or pharmaceutical industry to decide on the composition of a product that consists of multiple components. Before that some more investigations and refinements on the above smoothing technique have to be researched.

Two main angles could be the convexity of the smoothing function with respect to x and θ as well as the derivation of a good/best starting point for the problem at hand. One idea could be to use a high ϵ that could result in a convex or quasi-convex problem. Since oversmoothing in the presented approach unfortunately does not necessarily lead to such a problem (e.g. the higher the ϵ the more scenarios afar from the V@R are included). Some uncertainty on the feasibility of such an approach is owed to the fact that in the derivation of the AV@R Shapiro/Dentcheva/Ruszczynski (2009, p. 257ff) show that "[...] from the considered point of view, [the $AV@R_\alpha[losses] \leq 0$ condition] is the best convex conservative approximation of the chance constraint $[P(losses \leq 0) \geq 1 - \alpha]$". When it comes to a good starting point also some more work appears to be necessary. In this work - as well in the below mentioned article - the optimal AV@R composition values where used as a starting point for the smV@R. This is not a good approach since the location of the optimal V@R and AV@R may be far appart (e.g. if the optimization set is "star"-shaped). Especially in higher dimensional cases this is rather likely.

Finally, close to the completion of this thesis an article ("A Gradual Non-Convexation Method for Minimizing VaR" from 2012) was found where the authors considered a similar approach to smoothing the V@R. In their work they also included a feature that allowed to adapt ϵ (in their article it is called ρ) during the run of the NLP algorithm. Despite that it should be mentioned that the above was developed completely independently.

Literature

Cario/Nelson (1997): Modelling and Generating Random Vectors with Arbitrary Marginal Distributions and Correlation Matrix. Technique Report, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL.

Griva/Nash/Sofer (2009): Linear and nonlinear optimization.

Gaivoronski/Pfug (2005): Value-at-Risk in Portfolio Optimization: Properties and Computational Approach. Journal of Risk, Vol. 7, No. 2, pp. 1-31, Winter 2004-2005

Krokhmal/Palmquist/Uryasev (2001): Portfolio Optimization with Conditional Value-at-Risk Objective and Constraints.

Markowitz (1952): Portfolio Selection. The Journal of Finance, Vol. 7, No. 1. (Mar., 1952), pp. 77-91.

Niavarani/Smith (2012): Modeling and Generating Multi-Variate-Attribute Random Vectors Using a New Simulation Method Combined with NORTA Algorithm. Journal of Uncertain Systems Vol.7, No.2, pp.83-91, 2013

Pflug (2005b): Optimization of Non-standard Risk Functionals. Presentation by G. Pflug

Pflug/Römisch (2007): Modeling, Measuring and Managing Risk. World Scientific Publishing Co. Pte. Ltd.

Shapiro/Dentcheva/Ruszczynski (2009): Lectures on Stochastic Programming - Modelling and Theory. Society for Industrial and Applied Mathematics and the Mathematical Programming Society, First Edition

Uryasev et al. (2010): VaR vs. CVaR in Risk Management and Optimization. CARISMA Conference Presentation

Xi/Coleman/Li/Tayal (2012): A Gradual Non-Convexation Method for Minimizing VaR. based on Master Thesis by Jiong Xi, Weblinks: article: https://www.math.uwaterloo.ca/~tfcolema/articles/Convexation.pdf, thesis: https://uwspace.uwaterloo.ca/bitstream/handle/10012/6712/Xi_Jiong.pdf?sequence=1&isAllowed=y

Abstract

English

This thesis investigates the topic of portfolio selection and several approaches to it are discussed and compared.

In the first section the setup is presented accompanied by a short discussion on the currently most relevant approaches.

The next section covers and defines some attributes of risk measures and the later used risk measures are introduced. Since there already exist some standard methods for optimization approaches based on the AV@R and the standard deviation as risk criterion they are be used to compare and benchmark the smoothing approach to the V@R. For the latter - the V@R - no standard methods have become widely accepted yet and made it necessary to come up with some new ideas or refine and apply existing ones. The result - the smoothed V@R in the presented form - is the subject of the next section. In the first steps of this section a promising formulation for the optimization has been developed and introduced in detail. The core of this approach is to approximate the (empirical) cdf by a twice continously differentiable function that also is flexible when it comes to the choice of the mollifier - i.e. the approximation of the indicator function. Additionally their (first) derivate is given as closed form function and some challenges are discussed.

The final section is dedicated to show the applicability of the developed approach to a random and a real world example. This part of the thesis also covers - in part - the intricacies of the choice of non-linear optimization algorithms and corresponding parameters. After an adequate NLP algorithm is found the results of the smoothed V@R optimization is compared with the standard methods based on the standard deviation and the AV@R.

The conclusion finalizes this thesis and completes it with reflections on opportunities and ideas for further refinements of the presented method of the smoothed V@R.

Deutsch

Diese Arbeit untersucht das Gebiet der Portfolioselektion und diskutiert sowie vergleicht diverse Ansätze.

Im ersten Abschnitt wird das Setup beschrieben und von einer kurzen Präsentation der gegenwärtig relevantesten Ansätze begleitet.

Im nächsten Abschnitt werden wichtige Eigenschaften von Risikomaßen definiert und später benutzte Risikomaße erläutert. Da bereits Standard-Methoden zur optimalen Portfolioselektion basierend auf dem AV@R und der Standardabweichung existieren, werden diese als Vergleichsbasis verwendet und mit dem später erläuterten Ansatz - dem geglätteten (smoothed) V@R - verglichen. Für den V@R existieren bisher noch keine weiter verbreiteten Standardmethoden, woraus die Notwendigkeit entstand eigene Ideen zu entwickeln oder vorhandenen Ideen anzuwenden und weiter zu verfeinern. Der geglättete V@R in der vorliegenden Form ist das Ergebnis davon und wurde im danachfolgenden Kapitel genauer behandelt. Zuerst wurde eine vielversprechende Formulierung für die Optimierung entwickelt und danach noch detailliert aufbereitet. Zentral für diesen Ansatz ist die Annäherung der empirischen Verteilungsfunktion durch eine zweimal stetig ableitbare Funktion, die bezüglich der Glättungsfunktion zur Annäherung der Indikatorfunktion flexibel eingesetzt werden kann. Außerdem

wird die erste Ableitung dieser Approximation in geschlossener Form angegeben. Abschließend wurden offene Punkte und weiterentwicklungspotential aufgezeigt und diskutiert.

Im letzten Abschnitt wurde die Anwendbarkeit des entwickelten Ansatzes anhand eines Zufalls- und eines Real-Datensatzes demonstriert. In diesem Teil wurden die weiterführenden Herausforderungen bei der Auswahl eines adäquaten nicht-linearen Optimierungsalgorithmus behandelt und eine entsprechende Wahl bezüglich des Algorithmus und der notwendigen Parameter getroffen und begründet. Die Ergebnisse dieses Algorithmus wurden anschließend mit jenen der Standard-Methoden verglichen.

Abschließend wurden noch Überlegungen, weiterführende Möglichkeiten und Ideen diskutiert, die den präsentierten geglätteten V@R noch weiter verbessern könnten.