

Robust Portfolio Optimization

Daniel Bienstock

Columbia University, NYC

June 25 2007

The starting point: mean-variance optimization

$$\min \lambda \mathbf{x}^T \mathbf{Q} \mathbf{x} - \boldsymbol{\mu}^T \mathbf{x}$$

Subject to:

$$\mathbf{A} \mathbf{x} \geq \mathbf{b}$$

- $\boldsymbol{\mu}$ = expected asset returns, \mathbf{Q} = covariance of returns
- λ = risk aversion factor
- $\mathbf{A} \mathbf{x} \geq \mathbf{b}$ = portfolio construction constraints; e.g. country-sector constraints

→ How do we handle errors in \mathbf{Q} and/or $\boldsymbol{\mu}$?

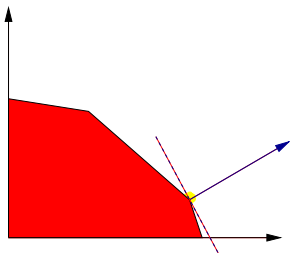
Mean-variance optimization is:

- Compelling from an intellectual standpoint; practical from a computational standpoint
- Very sensitive to data errors – may magnify perturbations
- Stability can be an issue

Popular approach: severely constrain asset weights – heavy handed?

Scenario I: Stability

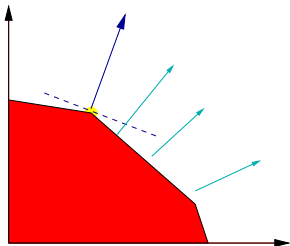
Data is fairly accurate, though possibly noisy – small errors are possible



→ Idiosyncratic decisions and small changes in data could have major impact

Scenario I: Stability

Data is fairly accurate, though possibly noisy – small errors are possible



→ Idiosyncratic decisions and small changes in data could have major impact

Scenario II: Hedging

Significant, but within order-of-magnitude, data uncertainty

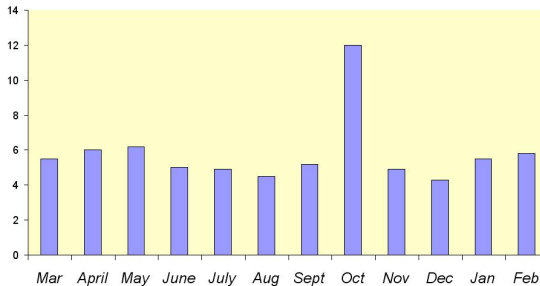
Example:

A certain parameter, α , is volatile. Its long-term average is **1.5** but it we could expect changes of the order of **.3**.

- Possibly more than just noise
- Could use deviations to our advantage, especially if there are several uncertain parameters that act “correlated”
- Are we guarding against risk or are we hedging?

Scenario III: Insurance

Real world data can exhibit undesirable and unexpected behavior



Classical goal: how can we protect without becoming too risk averse

Modeling Risk

- Geometry (structure) and scale (magnitude) are not the same thing
- Responsibility of a practitioner: to explore the risk-performance tradeoff
- Responsibility of an optimizer: to provide the tools that make the exploration possible
- And how about the risk that we do not model ...

Optimization under uncertainty

- Stochastic programming
- Adversarial queueing, online optimization
- “Risk-aware” optimization
- Optimization of utility functions as a substitute for handling infeasibilities

The classical stochastic programming approach

Suppose e.g. that returns are uncertain

- 1 Construct a **family of scenarios** describing possible outcomes for returns, with corresponding **probabilities**
- 2 Solve an appropriate optimization problem. Example: maximize expected return, or minimize value-at-risk

→ Widely used, well-understood from a theoretical and computational standpoint

Critiques on the stochastic programming approach

- 1 To achieve an accurate model we may need **many** scenarios
 - The scenarios are either explicitly constructed or sampled, but in either case the computational effort could prove very significant
 - We may not be able to achieve a satisfactory accuracy
- 2 Our choice of the probability distribution reveals our bias regarding the structure of uncertainty – we are exposed to structural risk

Robust Optimization

- Optimization under parameter (data) uncertainty
- Ben-Tal and Nemirovsky, El Ghaoui et al
- Uncertainty is modeled by assuming that data is not known precisely, and will instead lie in known sets.
- Example: a coefficient \mathbf{a}_i is uncertain. We allow $\mathbf{a}_i \in [l_i, u_i]$.
- Typically, a **minimization** problem becomes a **min-max** problem.

Example: Linear Programs with Row-Wise uncertainty

Ben-Tal and Nemirovsky, 1999

$$\min c^t x$$

Subject to:

$$Ax \geq b$$

→ The constraint matrix, A , is **uncertain**

→ The uncertainty is ill-defined

Problem to solve: find x

- with minimum $c^t x$,
- and such that $Ax \geq b$ for all A of interest

Example: Linear Programs with Row-Wise uncertainty

Ben-Tal and Nemirovsky, 1999

$$\min c^t x$$

Subject to:

$$Ax \geq b \quad \text{for all } A \in \mathcal{U}$$

\mathcal{U} = uncertainty set

→ the i^{th} row of A belongs to an ellipsoidal set \mathcal{E}_i

e.g. $\sum_j \alpha_{ij}^2 (\mathbf{a}_{ij} - \bar{\mathbf{a}}_{ij})^2 \leq 1$

→ can be solved using second-order cone programming techniques

Robust Linear Programs

$$\min c^t x$$

Subject to:

$$Ax \geq b$$

uncertainty in c

uncertainty in b

$$\min v$$

s.t.

$$Ax \geq b$$

$$v - c^t x \geq 0$$

$$\min c^t x$$

s.t.

$$Ax - yb = 0$$

$$y = 1$$

What is SOCP used for

→ Solving **conic** linear programs:

$$\min c^t x$$

Subject to:

$$Ax \geq b$$

$$x \in \mathcal{C}$$

Examples:

→ $\sum_{j=1}^n x_j^2 \leq 1$, a convex quadratic constraint

→ $\left(\sum_{j=1}^n x_j^2 \right)^{1/2} \leq yz$, a rotated cone constraint

→ $x^{3/2} \leq y$

How do we solve SOCPs

- Techniques derived from interior point methods for linear programming
- High-performance numerical linear algebra, theoretically efficient algorithms (polynomial time)
- Large, dense problems can prove challenging memory-wise and numerically
- Popular wisdom: SOCPs are “tractable”, as “easy as linear programs”
- The reality: linear programs are easier than quadratic programs, which are easier than SOCPs

Lobo and Boyd, 2000

Uncertainty in the covariance matrix \mathbf{Q} and the expected returns μ

- Ranges for the covariance entries: $\mathbf{Q}_{ij}^L \leq \mathbf{Q}_{ij} \leq \mathbf{Q}_{ij}^U$ and $\mathbf{Q} \succeq \mathbf{0}$
- Bound on the variance of a given portfolio: $L \leq \mathbf{w}^T \mathbf{Q} \mathbf{w} \leq U$
- Ellipsoidal uncertainty for μ : $(\mu - \bar{\mu})^T \mathbf{S} (\mu - \bar{\mu}) \leq 1$, $\mathbf{S} \succeq \mathbf{0}$

→ How do we get the \mathbf{Q}^L , \mathbf{Q}^U , L , U and \mathbf{S} ?

Lobo and Boyd, 2000

What problems they look at:

- Lowest variance problem:

$$\min_{\mathbf{x}} \sup_{\mathbf{Q}} \mathbf{x}^T \mathbf{Q} \mathbf{x}$$

- “Downside risk” problem:

$$\min_{\mathbf{x}} \sup_{\mathbf{Q}, \mu} \{ \lambda \mathbf{x}^T \mathbf{Q} \mathbf{x} - \mu^t \mathbf{x} \}$$

Lobo and Boyd, 2000

Algorithmics

- A mix of nonlinear, convex, and second-order cone programming
- The algorithms **may not** scale well with problem size
- Factor models: $\mathbf{Q} = \mathbf{V}^T \mathbf{F} \mathbf{V} + \mathbf{diag}$

(\mathbf{F} = “factor covariance matrix”)

it would be ideal to model uncertainty of \mathbf{V} directly, rather than \mathbf{Q}
 but e.g. variance bound on a given portfolio

$L \leq \mathbf{w}^T \mathbf{Q} \mathbf{w} \leq U$ is **nonconvex**

Goldfarb and Iyengar, 2002

Robust factor models

Classical factor model: $\mu = \bar{\mu} + V^T f + \epsilon$

- μ = (random) returns, $\bar{\mu}$ = expected returns,
- f = vector of (random) factors, V = factor loadings matrix
- ϵ : 0-mean normally distributed; etc
- So covariance matrix = $Q = VFV^t + D$
 F = factor covariance matrix; D = vector of variances of the ϵ_j

In other words, portfolio covariance =

$$\mathbf{x}^t \mathbf{Q} \mathbf{x} + \sum_j D_j x_j^2 = \mathbf{v}^t \mathbf{F} \mathbf{v} + \sum_j D_j x_j^2, \text{ where } \mathbf{v} = V^t \mathbf{x}.$$

Goldfarb and Iyengar, 2002

Uncertainty set

(Classical factor model: $\mu = \bar{\mu} + \mathbf{V}^T \mathbf{f} + \epsilon$)

- Assume that the factor covariance, $\mathbf{F} \succ \mathbf{0}$, is known
- All other data, in particular \mathbf{V} , belong to ellipsoidal sets:

$\mathbf{V} = \bar{\mathbf{V}} + \mathbf{W}$, where **each column** \mathbf{w}_i of \mathbf{W} satisfies

$$\mathbf{w}_i^T \mathbf{G} \mathbf{w}_i \leq \rho_i, \quad \mathbf{G} \succeq \mathbf{0}, \quad \rho_i \text{ given}$$

and

$$\bar{\mu}_j - \delta_j \leq \mu_j \leq \bar{\mu}_j + \delta_j \text{ for all assets } j$$

$$\mathbf{d}_j^L \leq \mathbf{d}_j \leq \mathbf{d}_j^U \text{ for all assets } j$$

Robust problems

- The min-max variance, with constraint on min-return, problem:

$$\min_{\mathbf{x}} \max_{V,D} \{ \mathbf{x}^t \mathbf{V} \mathbf{F} \mathbf{V}^t \mathbf{x} + \mathbf{x}^t \mathbf{D} \mathbf{x} \}$$

Subject to:

$$\min_{\mu} \{ \mu^t \mathbf{x} \} \geq \omega \quad (\omega = \text{lower bnd on return})$$

$$\sum_j \mathbf{x}_j = \mathbf{1}, \quad \mathbf{x} \geq \mathbf{0}$$

Similarly: **max-min return** objective, with constraint on **max variance**

How to solve

$$\min_{\mathbf{x}} \max_{V,D} \{ \mathbf{x}^t \mathbf{V} \mathbf{F} \mathbf{V}^t \mathbf{x} + \mathbf{x}^t \mathbf{D} \mathbf{x} \}$$

Subject to:

$$\min_{\mu} \left\{ \sum_j \mu_j \mathbf{x} \right\} \geq \omega \quad (1)$$

$$\sum_j \mathbf{x}_j = \mathbf{1}, \quad \mathbf{x} \geq \mathbf{0}$$

→ For any j , $\mu_j^L \leq \mu_j \leq \mu_j^U$, and $\mathbf{x}_j \geq \mathbf{0}$, so (1) becomes

$$\sum_j \mu_j^L \mathbf{x}_j \geq \omega$$

→ For any j , $\mathbf{d}_j^L \leq \mathbf{d}_j \leq \mathbf{d}_j^U$, so the maximum $\mathbf{x}^t \mathbf{D} \mathbf{x}$ is

$$\sum_j \mathbf{d}_j^U \mathbf{x}_j^2$$

$$\min_{\mathbf{x}} \max_{\mathbf{V}} \{ \mathbf{x}^t \mathbf{V} \mathbf{F} \mathbf{V}^t \mathbf{x} \}$$

→ For a given $\hat{\mathbf{x}}$, the “adversary” wants to solve:

$$\max_{\mathbf{V}} \{ \hat{\mathbf{x}}^t \mathbf{V} \mathbf{F} \mathbf{V}^t \hat{\mathbf{x}} \}$$

Subject to:

$$\mathbf{V} = \bar{\mathbf{V}} + \mathbf{W}, \text{ where each column } \mathbf{W}_i \text{ satisfies}$$

$$(\mathbf{W}_i^t \mathbf{G} \mathbf{W}_i) \leq \rho_i$$

- $\mathbf{F} \succ \mathbf{0}$, so (by scaling) can assume $\mathbf{F} = \mathbf{I}$
- Get:

$$\max \|\mathbf{V}^t \hat{\mathbf{x}}\|^2$$

Subject to:

$$(\mathbf{V}_i - \bar{\mathbf{V}}_i)^t \mathbf{G} (\mathbf{V}_i - \bar{\mathbf{V}}_i) \leq \rho_i \quad \text{for each } i$$

Experiments

- Simulated data: **500** assets and **43** factors
- Real data: **43** assets and **10** factors, **500** assets and **8** factors,
- Robust portfolios lose “little” in average return in exchange for gaining a lot in the worst case
- Robust portfolios less susceptible to noisy data
- Superiority over traditional portfolios primarily with high-confidence estimations
- Simulated data: up to **2000** assets and up to **200** factors: running time smoothly grows to \sim **600** seconds

Critique

- Technical point. Method requires positive-definite factor covariance: $\mathbf{F} \succ \mathbf{0}$
But in practical problems (e.g. small-cap) \mathbf{F} can be rank deficient, or nearly so (small eigenvalues).
- Ellipsoidal sets are constructed using statistical techniques, e.g. bootstrapping

Essentially, we are constructing multi-dimensional confidence intervals

So we are protecting against **statistical estimation errors**, and not **structural risk**.

Ceria and Stubbs, 2006

Uncertainty set for returns:

- $(\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})^t \mathbf{Q}^{-1} (\boldsymbol{\mu} - \bar{\boldsymbol{\mu}}) \leq \rho$

($\boldsymbol{\mu}$ = returns, $\bar{\boldsymbol{\mu}}$ = expected returns, \mathbf{Q} = covariance matrix)

- “zero net alpha”: $\sum_j (\mu_j - \bar{\mu}_j) \mathbf{x}_j = \mathbf{0}$

Problem solved:

$$\max_{\mathbf{x}} \min_{\boldsymbol{\mu}} \sum_j (\bar{\mu}_j - \mu_j) \mathbf{x}_j$$

Subject to:

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} \leq \omega$$

$$\sum_j \mathbf{x}_j = \mathbf{1}, \quad \mathbf{x} \geq \mathbf{0}$$

Tütüncü and Koenig, 2003

Uncertainty set:

- 1 Ranges for individual returns: $\mu_j^L \leq \mu_j \leq \mu_j^U$, all j
- 2 Ranges for the covariance matrix: $q_{ij}^L \leq q_{ij} \leq q_{ij}^U$, all j
and $\mathbf{Q} \succeq \mathbf{0}$

Min-max problems solved using special-purpose nonlinear programming techniques

Recap

The above robust models all use smooth (e.g. ellipsoidal) uncertainty sets, why?

- So as to use techniques from convex programming
- In particular, so as to obtain polynomial-time algorithms (“tractable formulations”)
- Is this the right metric?
- How about structural risk?
- Stochastic programming: cutting-plane algorithms very effective when handling complex uncertainty

Linear Programs with Row-Wise uncertainty

Bertsimas and Sim, 2002

$$\max c^t x$$

Subject to:

$$Ax \leq b \quad \text{for all } A \in \mathcal{U}$$

$$x \geq 0$$

→ in every row i at most Γ_i coefficients can change:

$$\hat{a}_{ij} - \delta_{ij} \leq a_{ij} \leq \hat{a}_{ij} + \delta_{ij}$$

→ for all other coefficients: $a_{ij} = \hat{a}_{ij}$.

Problem: find x that is **always feasible**, of maximum objective value

Application: mean-variance problems with uncertain returns

$$\min \lambda \mathbf{x}^T \mathbf{Q} \mathbf{x} - \mu^T \mathbf{x}$$

Subject to:

$$\mathbf{A} \mathbf{x} \geq \mathbf{b}$$

Uncertainty model:

- $\bar{\mu}_j - \delta_j \leq \mu_j \leq \bar{\mu}_j + \delta_j$ for each j
($\bar{\mu}_j$ = expected return for asset j)
- **at most** Γ returns vary from the mean

$$\min \lambda \mathbf{x}^T \mathbf{Q} \mathbf{x} - r$$

Subject to:

$$r - \mu^T \mathbf{x} \leq 0$$

$$\mathbf{A} \mathbf{x} \geq \mathbf{b}$$

Bertsimas and Sim: the robust problem can be solved as a single (larger) quadratic program

However, in this case the model is **rather simple**

$$\min \lambda \mathbf{x}^T \mathbf{Q} \mathbf{x} - \boldsymbol{\mu}^T \mathbf{x}$$

Subject to:

$$\mathbf{A} \mathbf{x} \geq \mathbf{b}$$

Uncertainty model:

$\mu_j - \delta_j \leq \mu_j \leq \mu_j + \delta_j$ for each j
at most Γ returns vary from the mean

Given asset vector \mathbf{x} ,
 its worst-case return is obtained by reducing the returns of the Γ
 largest assets to their minimum value $(\mu_j - \delta_j)$.