US Patent Pending

Uncertainty-Penalized Portfolio Optimization

“BETTER DECISIONS THROUGH ARTICULATED IGNORANCE”

OR “YOU KNOW WHAT YOU DON’T KNOW – WHY NOT TELL THE COMPUTER?”

JUL 18, 2017 BOSTON

ANISH R. SHAH, CFA

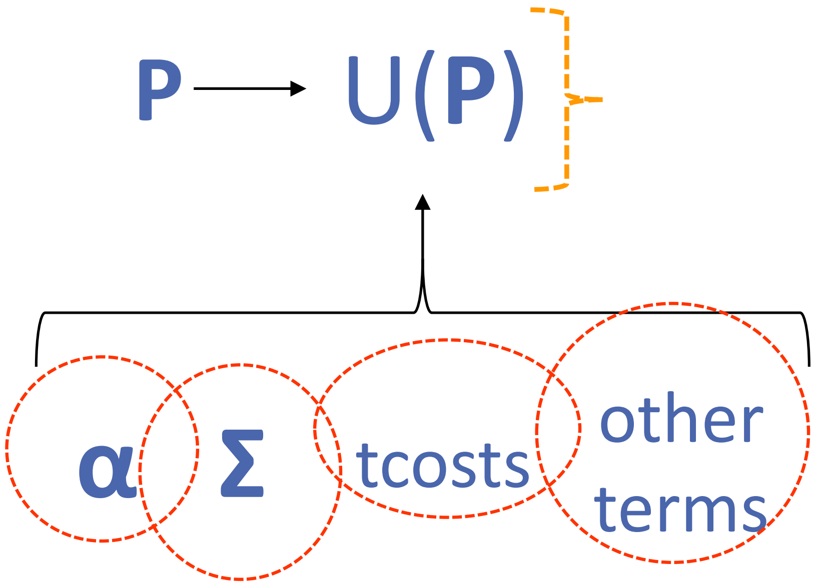
ANISHRS@INVESTMENTGRADEMODELING.COM

Outline – Ideas Old and New

A utility function evaluates portfolios [1952]

Inputs are inferred, inexact, uncertain [< 1980]

So, utility is uncertain [< 1980]



Explicitly estimate the uncertainty of the ingredients. e.g. mean and covariance via Bayesian estimation [old & new]

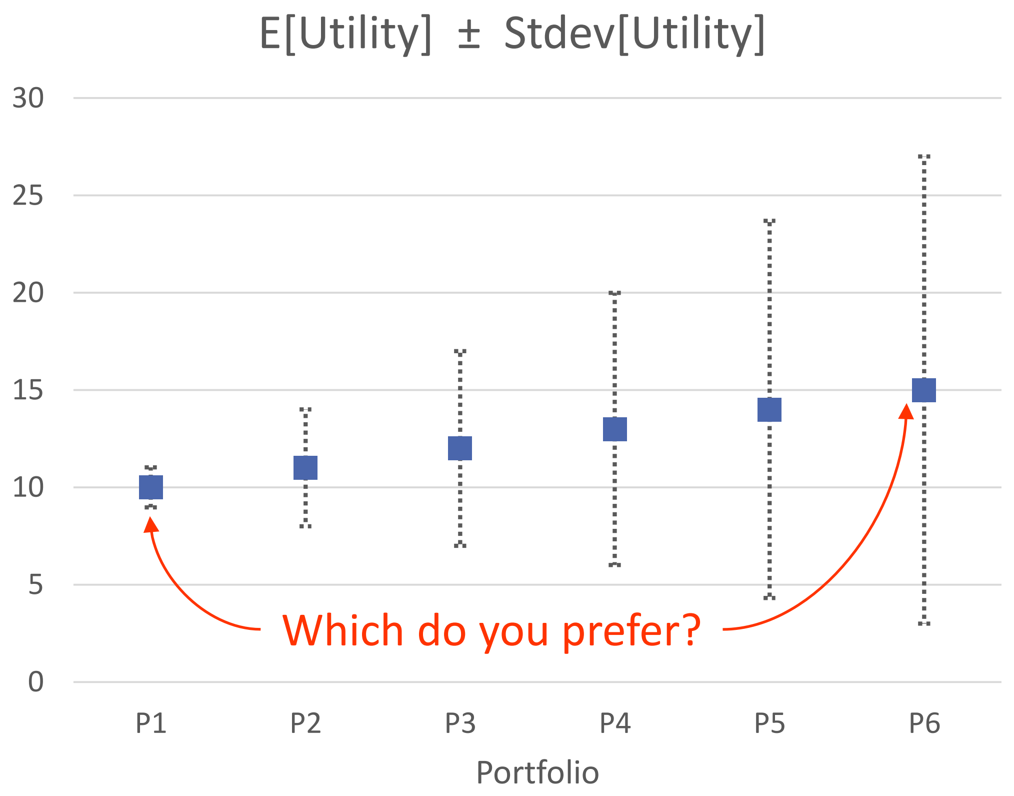
Smart parameterization yields mean and std dev of utility without simulation [new]

Outline (cont)

Can calculate E[Utility] and stdev[Utility] quickly for any arbitrary portfolio

Maximize E[Utility] - γ × stdev[Utility] to construct more reliable portfolios

For discussion: This works practically, but how to determine a particular investor's uncertainty aversion? Appears tied to rebalance frequency, turnover, taxes, …



(UPO) Uncertainty-Penalized Optimization Compared to Other Approaches

vs Michaud resampling (perturb input, average output)

- **UPO uses explicit estimates of uncertainty** – e.g. MSFT's beta is 1.1 ± 0.3, company Y's is 1.1 ± 0.9
  - Resampling – perturbing a point estimate – doesn't distinguish between the well and poorly predicted
- **UPO works with constraints and trading costs**; resampling doesn't

vs robust optimization (best worst case given a range on the inputs)

- **UPO operates on complicated utility functions** – multiple benchmarks, market impact costs, …
  - Robust optimization's max-min optimization requires particular structures
- **UPO doesn't guarantee finding a global max**; robust optimization does
  - Question in general: What's better – a richer model or a simplified model that has a guaranteed optimal solution?
- **UPO rolls uncertainty up to the utility level**
  - Can balance uncertainty and expected utility for different investor preferences. Unclear how to do so in robust optimization

Machinery

Getting Std Dev of Portfolio Utility

Utility is the sum of terms

- e.g. alpha, trading costs, tracking variances vs assorted benchmarks, …
- $U(\mathbf{w}) = \sum_k a_k\, g_k(\mathbf{w})$

Its variance is the sum of terms' variances and covariances

- $\mathrm{var}[U] = \sum_k a_k^2\, \mathrm{var}[g_k] + 2 \sum_{j<k} a_j\, a_k\, \mathrm{cov}[g_j, g_k]$

Assume what's not modeled is 0

- e.g. $\mathrm{var}[U] \approx \sum_k a_k^2\, \mathrm{var}[g_k]$ ignoring covariance between terms
- If you know and care about something, model it

Utility Terms: Uncertain Alpha

Conventional optimization takes an alpha vector

Uncertainty adds a covariance matrix

- Represents diversity of bets driving alpha

Conceptual fussiness vs robust optimization

- The robust optimization literature sometimes uses an estimate surrounded by an **elliptical confidence region, a frequentist idea**
- Here the estimate and dispersion are **Bayesian, "the expected value and covariance of alpha as I see it"**
- In practice, I doubt any difference – both estimated Bayesian

Utility Terms:
Uncertain Tracking Variance

Rather than being fixed, **the parts of a covariance model have estimates of mean and variance**

- Exposures, stock-specific variances, factor variances

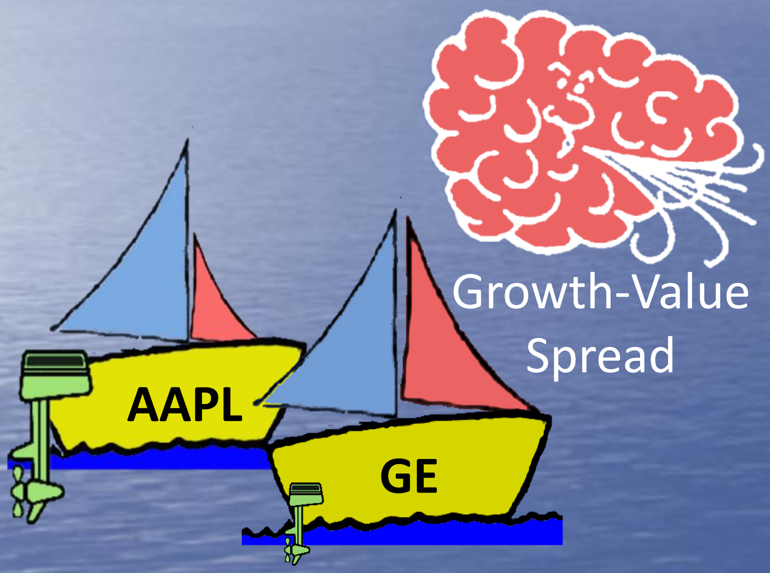
Mathematical details

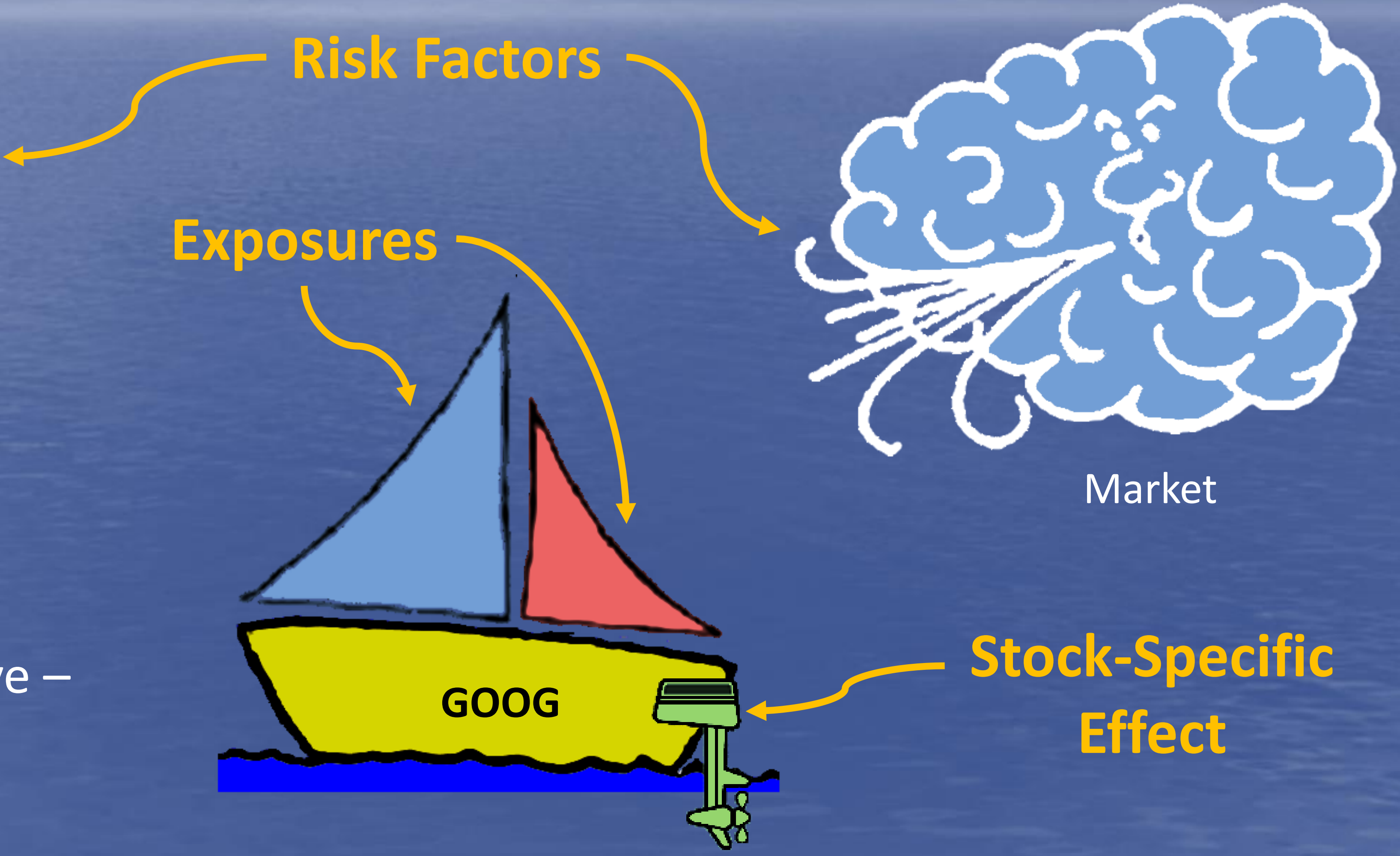
- Shah (2015) Uncertain Covariance Models, http://ssrn.com/abstract=2616109

Every conventional model (more or less) **can be rebuilt in its uncertain analog**

- **Tell your vendor you want one**
- A note: computation is done in an orthogonal representation, which for reports, is projected back to the original representation

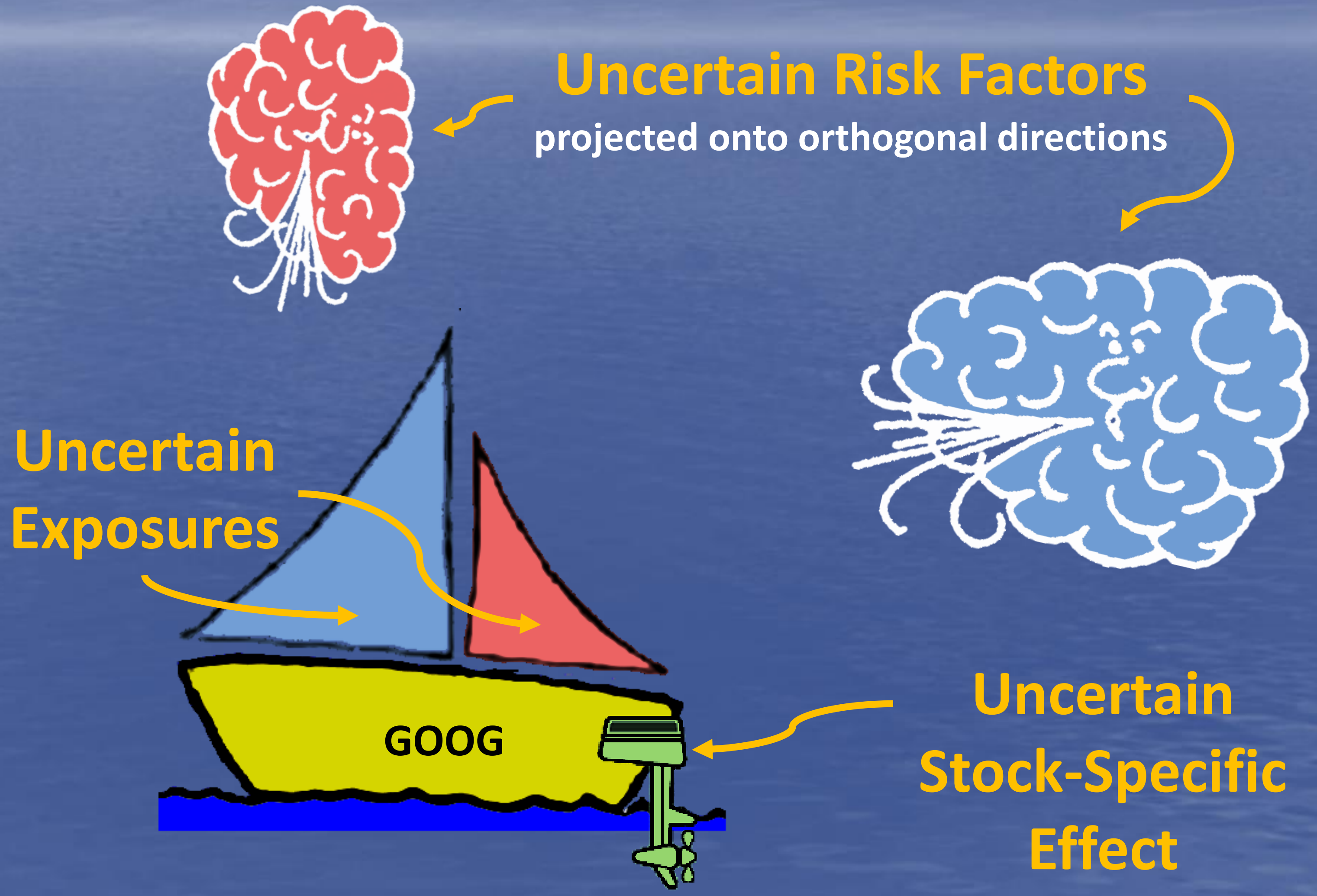
A Conventional Factor Covariance Model





Together these constitute a model of how securities move – jointly (winds and sails) and independently (motors)

An Uncertain Factor Covariance Model



Uncertain Exposures

Beliefs about future exposure to the factors are communicated as their mean and covariance

- $\hat{\boldsymbol{e}}_{GOOG} = \boldsymbol{E}[\boldsymbol{e}_{GOOG}]$ $\qquad \widehat{\boldsymbol{\Omega}}_{GOOG} = Cov[\boldsymbol{e}_{GOOG}]$
- Exposures can be correlated across securities
- Many ways to infer – Bayesian regression, Kalman filter, …

A portfolio's exposures are assumed Gaussian

- Needed for mathematical derivation of (uncertainty) variance of portfolio variance

Uncertain Factor Variances

Beliefs about the future factor variances are communicated as their mean and covariance

- Forecasts are the mean and covariance – according to uncertainty – of return variances
- Not Gaussian since variances ≥ 0

How the heck do you generate these?

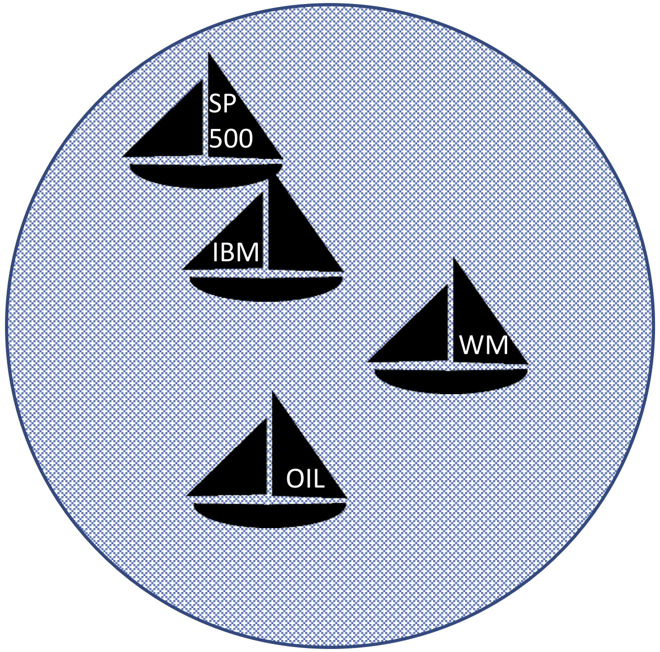
Shah (2014) Short-Term Risk and Adapting Covariance Models to Current Market Conditions

- http://ssrn.com/abstract=2501071

1. Forecast whatever you can, e.g. from VIX and cross-sectional returns, the volatility of S&P 500 daily returns over the next 3 months will be 25% ± 5% annualized
2. The states of quantities measured by the risk model imply a configuration of factor variances

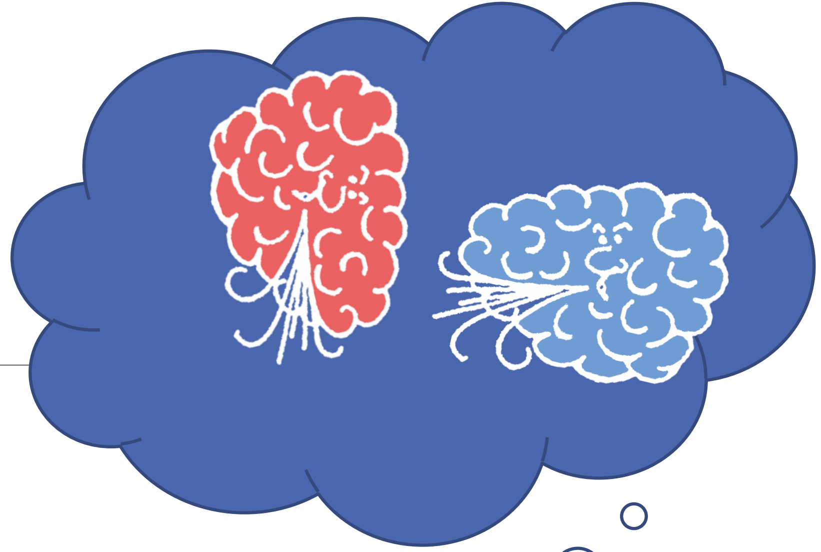
Since this inferred distribution of factor variances arises from predictions, it is a forecast

Modeling the Future by Adapting to Forecasts

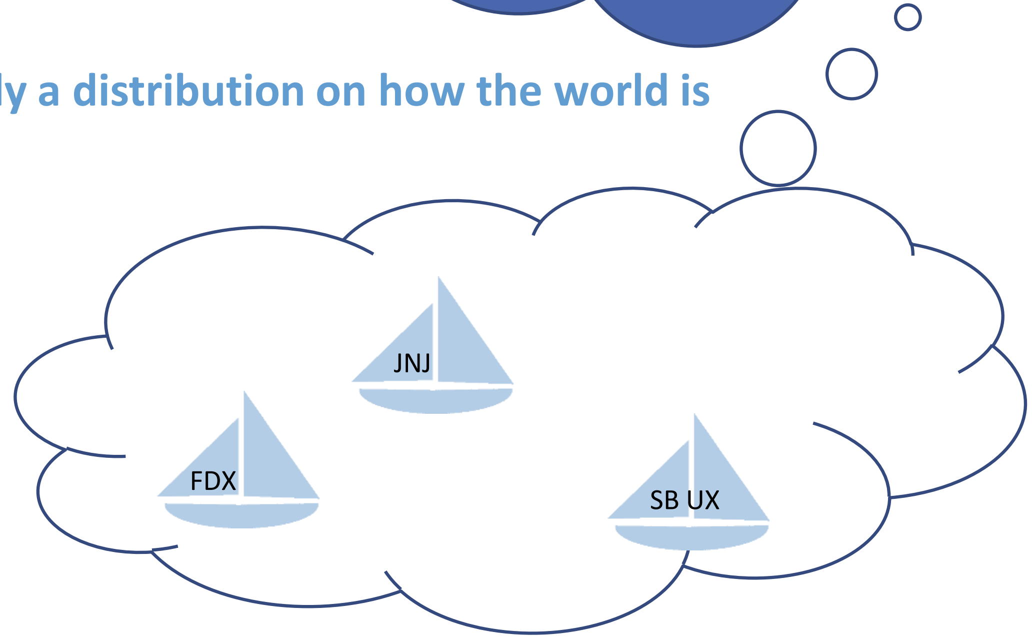


1. Noisy variance forecasts via all manner of Information sources

Implied vol, intraday price movement, news and other big data, …



2. Imply a distribution on how the world is



3. This extends to the behavior of other securities

All risk forecasts are improved

Utility Terms: Trading Costs & Others Terms

Optimization is done by a general purpose nonlinear optimization library

- Terms not restricted to a particular structure

For each term, need only

- Its E[ ] and var[ ] continuous and differentiable
- code that returns value and gradient of E[ ]
- code that returns value and gradient of var[ ]

Works with most market impact models

Optimization: Calculating Objective

$O(\mathbf{w}) = E[U(\mathbf{w})] - \gamma \text{ stdev}[U(\mathbf{w})]$ **the uncertainty-penalized utility objective**

To compute the objective and its gradient:

Recall utility is the sum of terms

- $E[U] = \sum_k a_k E[g_k]$
- $\text{var}[U] = \sum_k a_k^2 \text{ var}[g_k]$ (assuming not modeled between-term covariance is 0)

So, gradients are

- $\partial/\partial\mathbf{w}\ E[U] = \sum_k a_k\ \partial/\partial\mathbf{w}\ E[g_k]$
- $\partial/\partial\mathbf{w}\ \text{var}[U] = \sum_k a_k^2\ \partial/\partial\mathbf{w}\ \text{var}[g_k]$
- $\partial/\partial\mathbf{w}\ \text{stdev}[U] = \partial/\partial\mathbf{w}\ \text{var}[U] / [2\ \text{sqrt}(\text{var}[U])]$ (by chain rule)
- $\partial/\partial\mathbf{w}\ O = \partial/\partial\mathbf{w}\ E[U] - \gamma\ \partial/\partial\mathbf{w}\ \text{stdev}[U]$

Optimization Algorithm

Maximizing a non-concave objective, so local max need not be a global max

First solve a QP that globally resembles the structure

- Solution serves as initial guess for the harder optimization and verifies constraints' feasibility

To nonlinear optimization library, pass functions that calculate

- value and gradient of objective
- value and jacobian of constraints (each constraint is written as g[**w**] <= 0 or g[**w**]=0)

Easy to swap out engine and take advantage of new optimization technology

Uncertain
Portfolio Optimizer

IGM Uncertain Optimizer in R
XML or list input

Define data sources

```
...
<datasource>
  <name>proxy</name>
  <type>csv</type>
  <filename>proxy.csv</filename>
</datasource>
<datasource>
...
```

proxy.csv

bbgid	ticker	name	proxy1	proxy2
BBG000B9WH86	AA	ALCOA INC	RIO	BBG000B9XRY4
BBG000B9XRY4	AAPL	APPLE INC	MSFT	BBG000BPH459
…				

Optimization inputs

```
 ...
  <utilityterms>                <!-- can have multiple entries of any type of utility term -->
    <stdevpenalty>1</stdevpenalty>  <!-- penalty on uncertainty of utility, will maximize E[U] - penalty x stdev[U] -->
    <term>
      <type>variance</type>         <!-- relative variance vs a benchmark -->
      <scalar>-0.5</scalar>
      <model>cov_daily</model>      <!-- name of a covariance datasource -->
      <benchmark>bench2</benchmark> <!-- empty or name of benchmark defined above -->
      <unmatchedproxy>              <!-- proxy for securities not covered by model. Can have multiple unmatched proxy records -->
        <value>proxy$proxy1</value> <!-- field containing security id -->
        <type>TICKER</type>         <!-- type of security id -->
      </unmatchedproxy>
    </term>
    ...
```

> library(IGMOptimizer)

Read datasources and map by id

```
ds <- IGMreadDataSources(file="dsmap.xml")
```

Read optimization settings and create data structures

```
optdat <- IGMparseOptXML(file="optsettings.xml", ds)
```

Maximize globally similar QP for initial guess and to verify feasibility of constraints

```
w0 <- IGMoptimizeQPApproximation(dat, upenalty)
```

Maximize true objective

```
opt <- IGMoptimizePortfolio(w0, optdat, upenalty)
```

Investment Grade Modeling LLC

High-tech boutique modeling, in particular, for situations where accuracy has a premium

- Leverage
- Firm managing its own capital

Risk models are based on original research

- Shah, Anish (2014) Short-Term Risk and Adapting Covariance Models to Current Market Conditions
- Shah, Anish (2015) Uncertain Covariance Models

Patent pending uncertain utility optimization

- Maximizes uncertainty-penalized utility. Avoids being burned by bad numbers
- R library

ANISH R. SHAH, CFA
ANISHRS@
INVESTMENTGRADEMODELING.COM