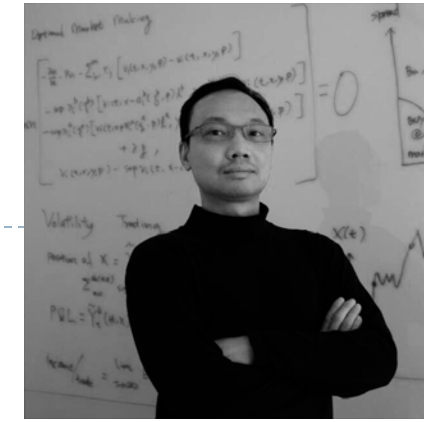
资产配置技术的新突破

李克辛
haksun.li@numericalmethod.com
www.numericalmethod.com

演讲者介绍



- 李克辛　教授
- NM LTD., 行政总裁
- 复旦—中植大数据金融与投资研究院　副院长
- 多家大学的兼职教授，曾任教或兼职任教于新加坡国立大学、新加坡海洋理工大学、复旦大学以及香港科技大学等多所高校
- 曾在法国巴黎银行、瑞银集团担任量化交易员和分析师
- 芝加哥大学纯数学学士、金融数学硕士
- 密歇根大学人工智能、计算机科学与工程硕士、博士

阿尔法策略在中国市场的应用情况概述

举例说明中国市场中阿尔法策略构建步骤

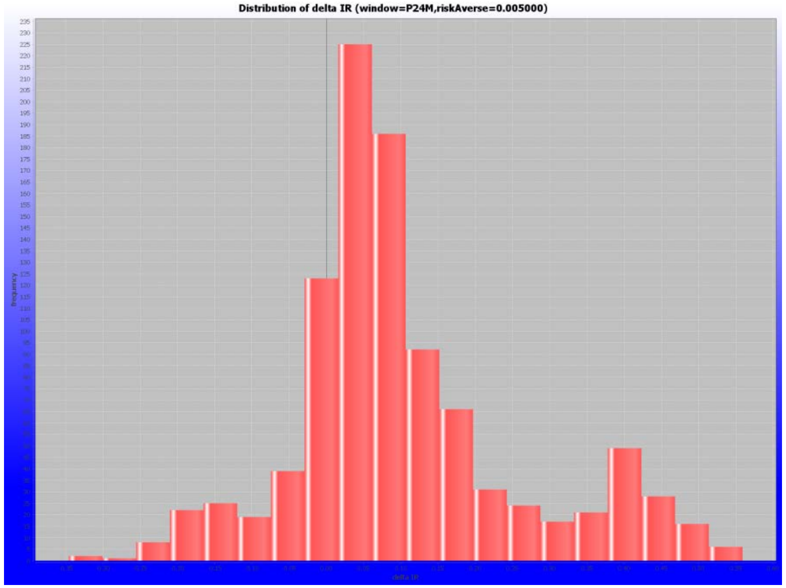
- 从1000个因子中选择一部分因子组成因子库
- 计算每个因子的信息比率
- 计算每个因子的信息比率占整个筛选出的因子库信息比率的权重
- 结合因子库的价值给股票打分
- 根据分值从每个行业选择股票
- 从各行业中选出分值高的前20%股票
- 安排每个行业在整个市场中的比重
- 安排每个股票在各行业中的比重
- 用中证800指数进行对冲

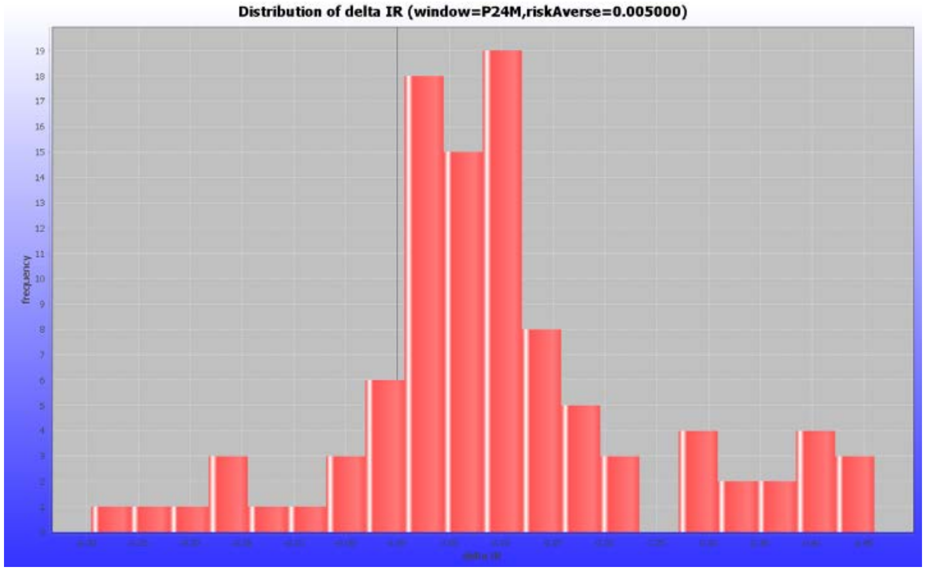
阿尔法策略在中国市场应用中的问题

- 阿尔法的失败原因
  - 市场特性的变化，市值因子、规模因子、大小盘风格逆转
  - 负基差，基差的不可预测性
- 大同小异
  - 策略大同小异
  - 因子大同小异
  - 权重配置大同小异
- 因子利用只是股票的选择和过滤
  - 没有预测功能
- 需说是量化，但当中并没有数学模型
  - 只是一些试探性的

解决方案—优化资产配置

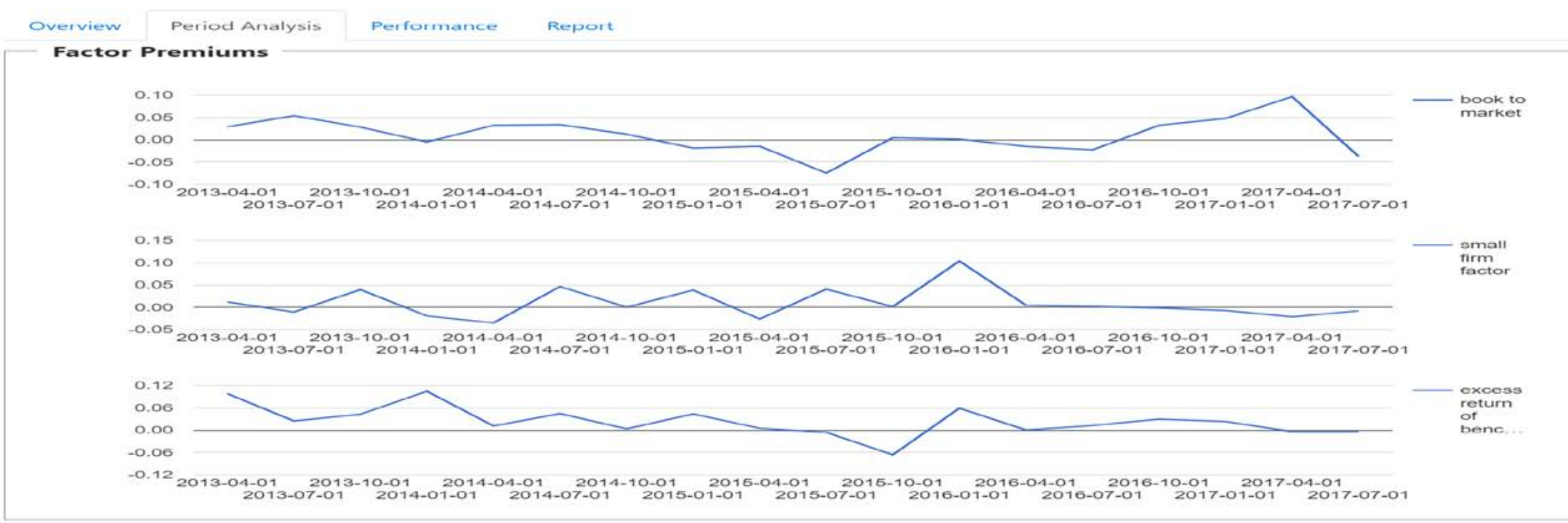
- 假设同样组合的股票在较长期限后，由于不同的权重配比会产生不同的损益结果。





解决方案—预测因子模型

- 我们可以用因子构建数学预测模型
  - 模型预测股票收益
  - 模型不再是个简单的股票筛选过滤器
- 可以科学地计算出因子的有效性、稳定性以及不同时间段因子的特征



马可威茨优化投资组合中的问题

如何把资产组合最优化

- FOF资产配置
  - 每个基金中放多少资金？
- 证券投资组合资产配置
  - 每个策略中放多少资金？
- 阿尔法策略资产配置
  - 每只股票放多少资金？

马可威茨

- 1952年，马可威茨首次提出将资产配置作为一个数学问题
  - 这个理论被写在了各大高校的MBA课程的教科书里
- 马可威茨上述理论在1990年获得了诺贝尔经济学奖

现代资产组合理论

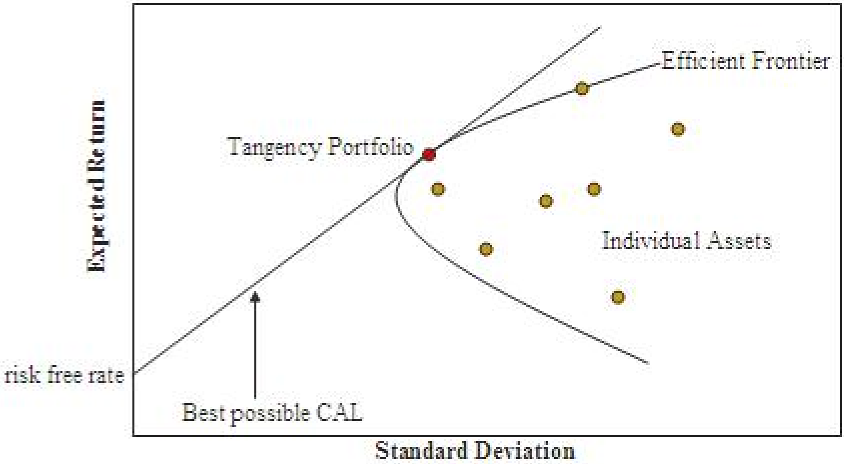
- 单个资产的风险和回报应该通过它对投资组合整体风险和回报的贡献来评估，而不是由它自己的风险和回报来决定的。
- 优化的均值方差。
- 投资者是厌恶风险的，这意味着，如果两个投资组合提供相同的预期收益，投资者就会选择风险较小的投资组合。
- 想要获得更高回报的投资者必须接受更多的风险。
- 投资者可以根据风险（容忍）参数找到自己的风险规避特性。

现代资产组合理论—用数学理论阐述

- $\max_{\omega}\{\omega\,\mathrm{E}(r_{t+1}) - \lambda\omega'\Sigma_t\omega\}$
  - ω 为最优组合的权重
  - $\mathrm{E}(r_{t+1})$ 为下一期的期望收益
  - Σ_t 为资产的协方差矩阵
- 限制条件: $A\omega \leq b$
  - 不允许卖空: $-\mathrm{I}\omega \leq 0$
- 或者
  - $\min_{\omega}\{\omega'\Sigma_t\omega - \lambda\omega\,\mathrm{E}(r_{t+1})\}$
- 解决方案: 二次规划（ Quadratic Programming ）
- NM:
  - http://redmine.numericalmethod.com/projects/public/repository/svn-algoquant/show/core/src/main/java/com/numericalmethod/algoquant/model/portfoliooptimization/markowitz

有效边界

- 假设
  - $\omega\,\mathrm{E}(r_{t+1}) = \mu$
- 求解 ωs.t.,
  - $\omega_{eff} = \underset{\omega}{\operatorname{argmin}}\{\lambda\omega'\Sigma_t\omega\}$



马可威茨理论的问题

- 需要知道均值和协方差。
  - 需要估计的参数数量过多: $N + \frac{N^2+N}{2}$.
    - 当 $N = 300$ 时, 我们需要估计45,450 个参数。
    - 当 $N = 3000$ 时, 我们需要估计4,504,500 个参数。
    - Chopra &Ziemba(1993) 的研究显示均值中误差的重要性是方差中误差的十倍，方差中误差的重要性是写方差中误差的2倍。
  - 时间是不断变化的。受商业周期影响。

取样矩形协方差的问题

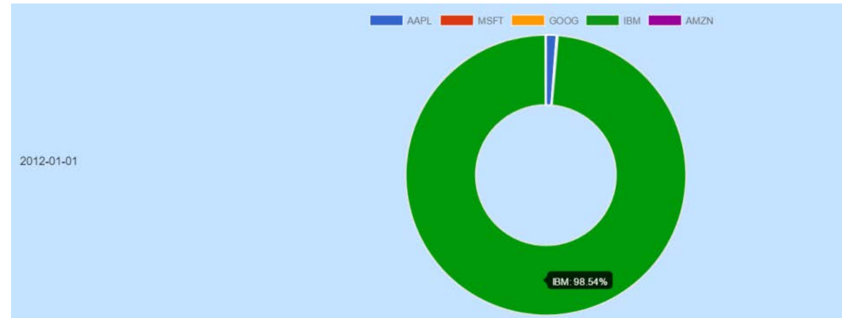
- 样本协方差矩阵通常是病态的，几乎是奇异的，有时甚至不是可逆的，有时甚至不是正半定的。
  - 维度: p,
  - 样本数量: n
  - $\frac{p}{n} > 1$, 矩阵不可逆
  - $\frac{p}{n} < 1$ 是不可微的, 矩阵是病态的
- 股票间的线性相关性。
  - 异步数据
  - 不完整数据
  - 压力测试引起的人为变化
- 误差的最大化:
  - 最大样本特征值是系统性地上偏。
  - 最小样本特征值是系统性地下偏。
  - 反转样本协方差矩阵会显著增加估计误差。
  - 资金被分配到了最不可靠的极端特征值上。

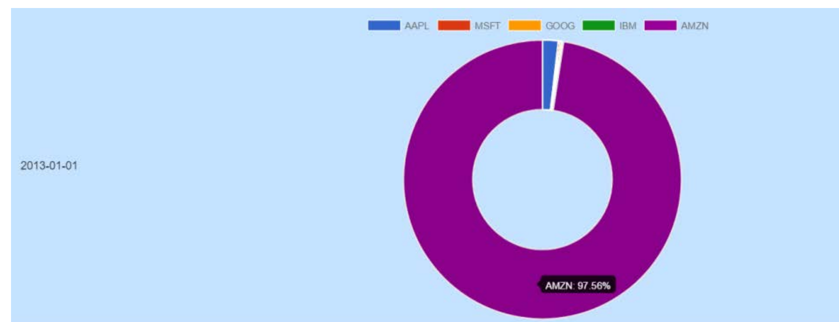
取样均值的问题

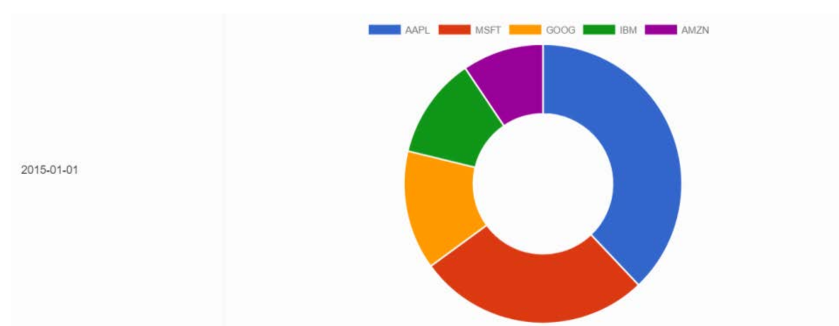
- 样本均值只是对两个数据点（即两个端点）的估计，它忽略了样本的大小。
- 给定一组历史收益率$\{r_1, \ldots, r_t\}$, 样本均值是：
- $\bar{r} = \sum_{i=1}^{t} r_i$
- $\approx \sum_{i=1}^{t} \log(1 + r_i) = \sum_{i=1}^{t} \log(p_i) - \log(p_{i-1})$
- $= \log(p_t) - \log(p_0)$
- 假设收益率服从高斯分布。
  - Nassim Nicholas Taleb认为:股票市场崩溃后（1987年），诺贝尔奖委员会表彰了两个理论学家，Harry Markowitz和William Sharpe，他们以高斯假设为前提，构建了漂亮的柏拉图模型，并发展成为现代投资组合理论。简单地说，如果去掉了高斯假设并将价格视为可伸缩的，那么该理论剩下的只是一纸空文。诺贝尔奖委员会本可以测试夏普和马科维茨的模型——这些模型犹如在互联网上销售的狗皮膏药——但是斯德哥尔摩的委员们似乎没有一个人这么考虑过。

多样化的问题

- Litterman & et al. (1992, 1999, 2003):
  - 当资金不受约束时，投资组合可以有大量的长期和短期头寸。
  - 当资金受制于长期投资约束时，这些长期投资资金只能分配给少数资产。
- Best & Grauer (1991):预期收益的小幅增加可能消耗掉一半的资本。





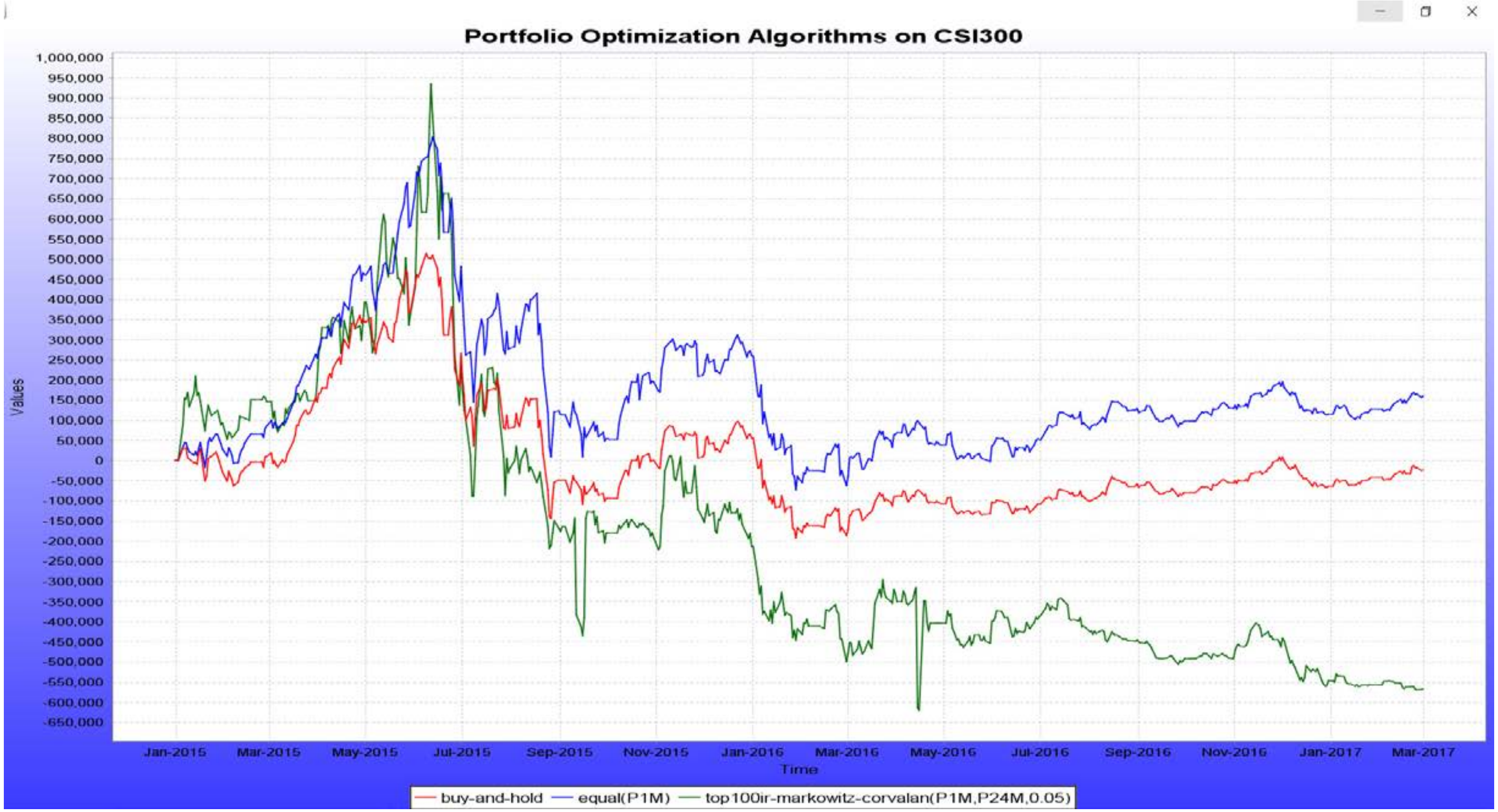


关于一些限制的问题

- 最小化方差
  - $\max_{\omega} \omega \, \mathrm{E}(r_{t+1})$, s.t.,
  - $\omega' \Sigma_t \omega \leq \sigma_{MAX}$
  - $1'\omega = 1$
  - $\omega \geq 0$
- 市场冲击
  - $\max_{\omega} \left\{ \omega \, \mathrm{E}(r_{t+1}) - \lambda_P \omega' \Sigma_t \omega - \lambda_M \sum_{j=1}^{n} \left(m_j |\omega_j|^{\frac{3}{2}} + c_j |\omega_j| \right) \right\}$
- 多样性限制（行业暴露）
  - $\sum_{j \in S_i} \left| \omega_j^0 + \omega_j \right| \leq u_i$ for 行业 $i = 1, \dots, S$
- 税收、交易成本、等等.

关于业绩表现的问题

- 整个周期的损益通常比1／N阶段的损益表现差（权重相等）。



关于马可威茨理论的评述

- Wesley Gray：马科维茨赢得诺贝尔奖，这是基于他用数学方法确定均值方差有效组合。当他把整个想法用到这个真实世界的时候，有意思的事情发生了：均值方法表现并并不佳。在考虑资产配置的最优化方法时，这个诺贝尔奖的想法对投资者来说是没有价值的补充。虽然复杂，但没有价值。

实用投资组合优化的求解方法

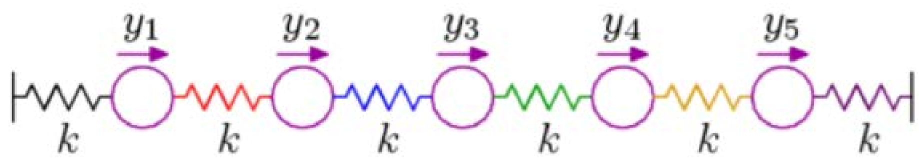
估计协方差的解决方法——降维

- 通过多因子模型降低维度
  - 将第i个资产收益率r_i与k个因子$f_1, \ldots, f_k$相关量：
  - $r_i = \alpha_i + (f_1, \ldots, f_k)'\beta_i + \epsilon_i$
  - α_i, β_i 是未知的回归参数; ϵ_i 是不可观测的随机噪声，均值为零且不相关。
  - $\mathrm{Cov}\left(r_{it}, r_{jt}\right) = \beta_{it}' \, \mathrm{V}(f) \, \beta_{jt}' + \mathrm{Cov}\left(\epsilon_{it}, \epsilon_{jt}\right)$
  - E.g., alpha 策略, Fama-French 模型, CAPM, APT
  - NM:
    - http://redmine.numericalmethod.com/projects/public/repository/svn-algoquant/show/core/src/main/java/com/numericalmethod/algoquant/model/factormodel

估计协方差的解决办法——收缩估计

- 将极值特征值拉回平均值。
- Ledoit and Wolf (2003, 2004):
  - $\hat{\Sigma} = \hat{\delta}\hat{F} + (1 - \hat{\delta})S$
    - $\hat{\delta}$ 是最有收缩常数的估计值
    - $\hat{F}$ 由先验分布的均值或者结构化的协方差矩阵表示，后者的参数数量远小于$N + \frac{N^2+N}{2}$.
    - S 是样本协方差
  - NM:
    - http://www.numericalmethod.com/javadoc/suanshu/com/numericalmethod/suanshu/stats/descriptive/covariance/LedoitWolf2004.html
    - http://www.numericalmethod.com/javadoc/suanshu/com/numericalmethod/suanshu/model/returns/moments/MomentsEstimatorLedoitWolf.html
- Ledoit and Wolf (2012): 非线性收缩

逆协方差矩阵VS协方差矩阵



inverse-covariance matrix or covariance matrix?

$$\mathbf{K}^{-1} = \frac{k}{T}\begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \qquad \mathbf{K} = \frac{T}{k}\begin{bmatrix} 0.83 & 0.67 & 0.50 & 0.33 & 0.17 \\ 0.67 & 1.33 & 1.00 & 0.67 & 0.33 \\ 0.50 & 1.00 & 1.50 & 1.00 & 0.50 \\ 0.33 & 0.67 & 1.00 & 1.33 & 0.67 \\ 0.17 & 0.33 & 0.50 & 0.67 & 0.83 \end{bmatrix}$$

估算协方差的解决方案——协方差的选择

- Dempster (1972):通过将协方差矩阵的逆元素设置为零，可以简化多元正态总体的协方差结构。
- Awoye, OA; (2016): Graphical LASSO
- NM:
  - http://www.numericalmethod.com/javadoc/suanshu/com/numericalmethod/suanshu/model/covarianceselection/lasso/CovarianceSelectionGLASSOFAST.html
  - http://www.numericalmethod.com/javadoc/suanshu/com/numericalmethod/suanshu/model/covarianceselection/lasso/CovarianceSelectionLASSO.html

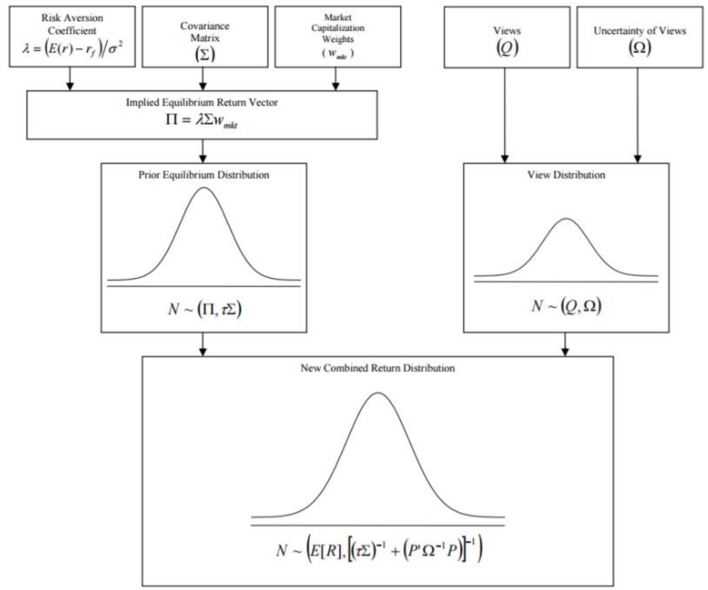
估算协方差的方法——正定矩阵

- 正定矩阵
- Goldfeld, Quandt and Trotter
- Matthews and Davies
- 正对角
- NM:
- http://www.numericalmethod.com/javadoc/suanshu/com/numericalmethod/suanshu/algebra/linear/matrix/doubles/operation/positivedefinite/package-summary.html
- 协方差矩阵
- Nicholas J. Higham (1988, 2013)
- Defeng, Sun (2011, 2006)

估计均值的解决办法——统计学的方法

- 交易信号
  - NM:
    - http://redmine.numericalmethod.com/projects/public/repository/svn-algoquant/show/core/src/main/java/com/numericalmethod/algoquant/model
- 多因子模型: $r_i = \alpha_i + (f_1, \ldots, f_k)'\beta_i + \epsilon_i$
- 缩减

估计均值的方法—Black-Litterman



- 组合的收益率向量
  - $\mathrm{E}(R) = [(\tau\Sigma)^{-1} + P'\Omega^{-1}P]^{-1}[(\tau\Sigma)^{-1}\Pi + P'\Omega^{-1}Q]$
    - P: 一个识别符合观点中资产的矩阵 $(K \times N)$
    - Ω: 以表达观点的对角协方差矩阵误差项，代表了每个观点的不确定性 $(K \times K)$
    - Π: 隐含的均衡收益率向量 $(N \times 1)$
    - Q: 观点向量$(K \times 1)$

避免过度多样化—使用限制量

- Black-Litterman
- 提出采用一些避免多样化的限制，如：
  上下界值

多样化的解决方案—几乎是有效的证券投资组合

- 均值-方差优化法（ MVO ）是从风险—报酬的角度给出一个最优投资组合。
- 均值-方差优化法无意去追求一个分散的投资组合
- 有效边界上的许多投资组合确实很集中。
- 然而，在有效边界的小范围周边有很多多样化的投资组合。
- 几乎是有效的投资组合：
  - $\max_{\omega} D(\omega)$ s.t., (D 是多样化标准)
    - $\sqrt{\omega'\Sigma\omega} \leq \sigma^{\text{eff}} + \Delta\sigma$, 组合方差标准放宽
    - $R^{\text{eff}} - \Delta R \leq \omega' r$, 组合期望收益标准放宽
    - $1'\omega = 1$
- NM:
  - http://numericalmethod.com/blog/2013/06/19/solving-the-corner-solution-problem-of-portfolio-optimization/
  - http://www.numericalmethod.com/javadoc/suanshu/com/numericalmethod/suanshu/model/corvalan2005/diversification/package-summary.html

二次锥规划

- $\min_x f'x$, s.t.,
  - $\|A_i x + b_i\|_2 \leq c_i' x + d_i,\ i = 1, \ldots, m$
  - $Fx = g$
- LP, QP
- 解决方案: 内点法（ interior point method ）

加入限制条件的解决方案—二阶锥规划

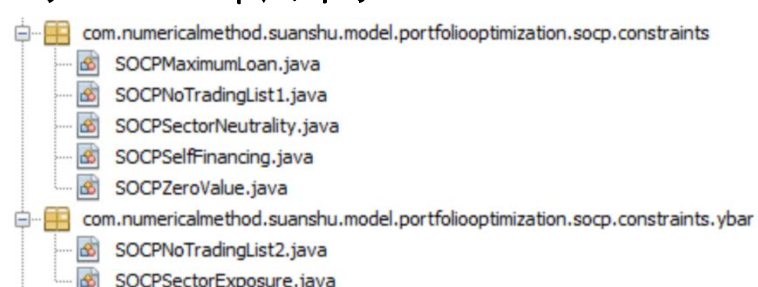
- 市场冲击
  - $\sum_{j=1}^{n}\left(m_j|\omega_j|^{\frac{3}{2}}\right) \le t_2$

$$\|0\|_2 \le t_2 - \sum_{j=1}^{n} \beta_j,$$
$$\|0\|_2 \le \bar{x}_j - (y_j - w_j^0),\ j = 1, \cdots, n,$$
$$\|0\|_2 \le \bar{x}_j - (-y_j + w_j^0),\ j = 1, \cdots, n,$$
$$\left\| \begin{pmatrix} \bar{x}_j \\ \frac{\beta_j}{m_j} - \frac{s_j}{2} \end{pmatrix} \right\|_2 \le \frac{\beta_j}{m_j} + \frac{s_j}{2},\ j = 1, \cdots, n,$$
$$\left\| \begin{pmatrix} s_j \\ \frac{1-\bar{x}_j}{2} \end{pmatrix} \right\|_2 \le \frac{1+\bar{x}_j}{2},\ j = 1, \cdots, n.$$

- 多样化显示（行业暴露）$\sum_{j\in S_i}\left|\omega_j^0 + \omega_j\right| \le u_i$ for sector $i = 1, \ldots, S$

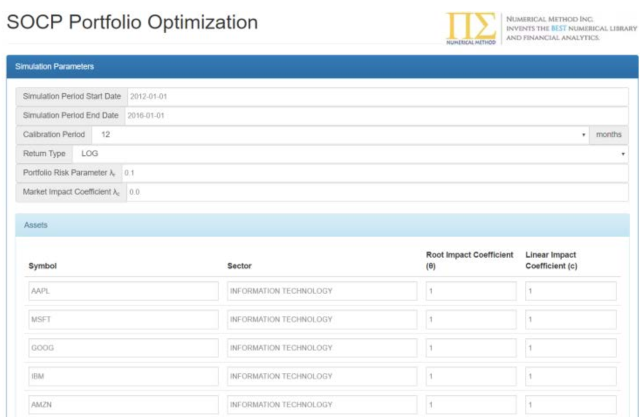
$$\|0\|_2 \le -\sum_{j\in S_i} \bar{y}_j + u_i \Longleftrightarrow \|A_i^\top z + C_i\|_2 \le b_i^\top z + d_i, \quad i = 1, \cdots, k$$
$$A_i^\top = 0_{1\times n},\ C_i = 0,\ b_i = -\sum_{j\in S_i} e_j,\ d_i = u_i,\ z = \bar{y},$$

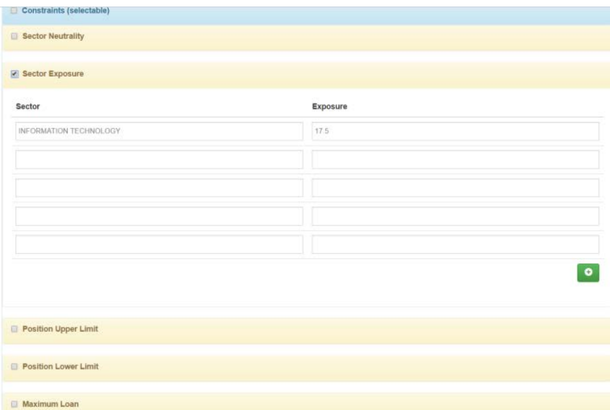
- 很多其他的限制可以被模型化为SPCP限制。
- NM 有很多这样的限制.

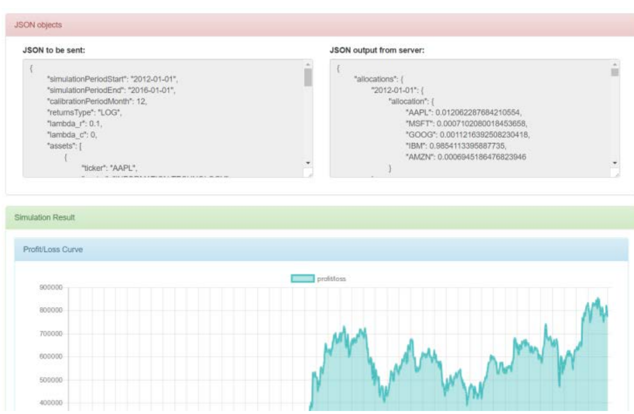


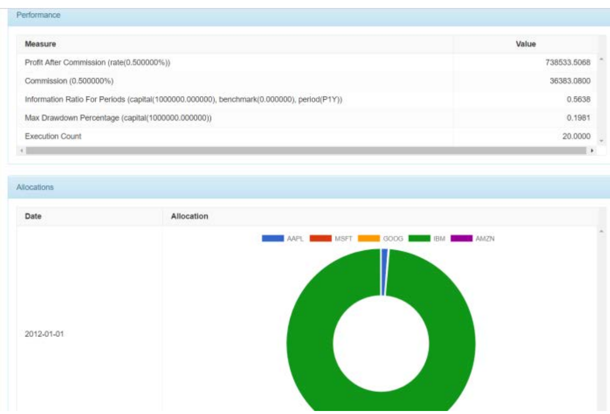
NM的二阶锥规划的优化程序

- https://sscloud-201608.appspot.com/socp-portfolio-optim.html



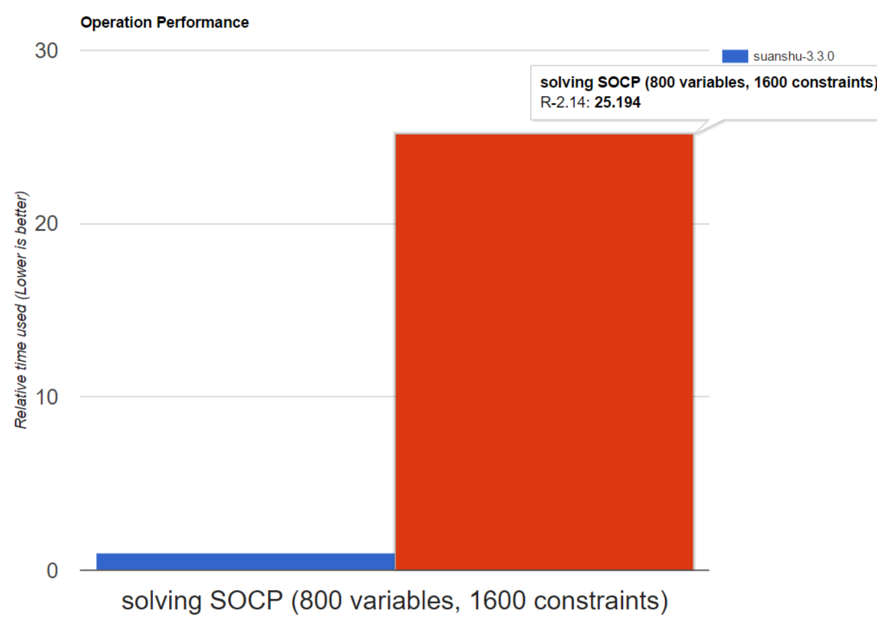






二阶锥规划的优化程序

- NM公司的优化程序
- 　比随意优化快25倍
- MOSEK数学优化软件包
- Gurobi数学规划服务器
- CPLEX
- XPRESS



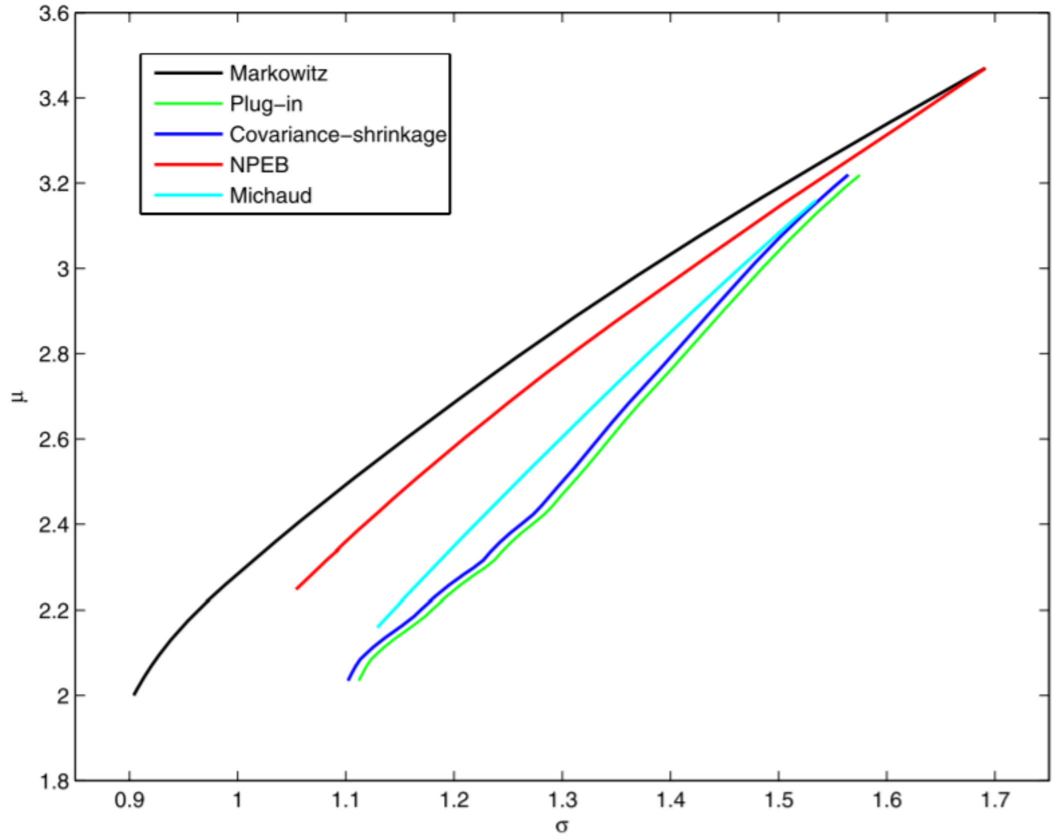
性能解决方案—优化估计

- 我们将NM公司的模块和算法结合起来做成更好的均值-方差优化模型。
- 优化均值估算
- 优化协方差估算
- 优化限量模型
- 优化分化标准
- NM公司均值-方差优化模型的比较框架：
  - http://redmine.numericalmethod.com/projects/public/repository/svn-algoquant/show/core/src/main/java/com/numericalmethod/algoquant/model/portfoliooptimization/simulation

未知均值和未知协方差

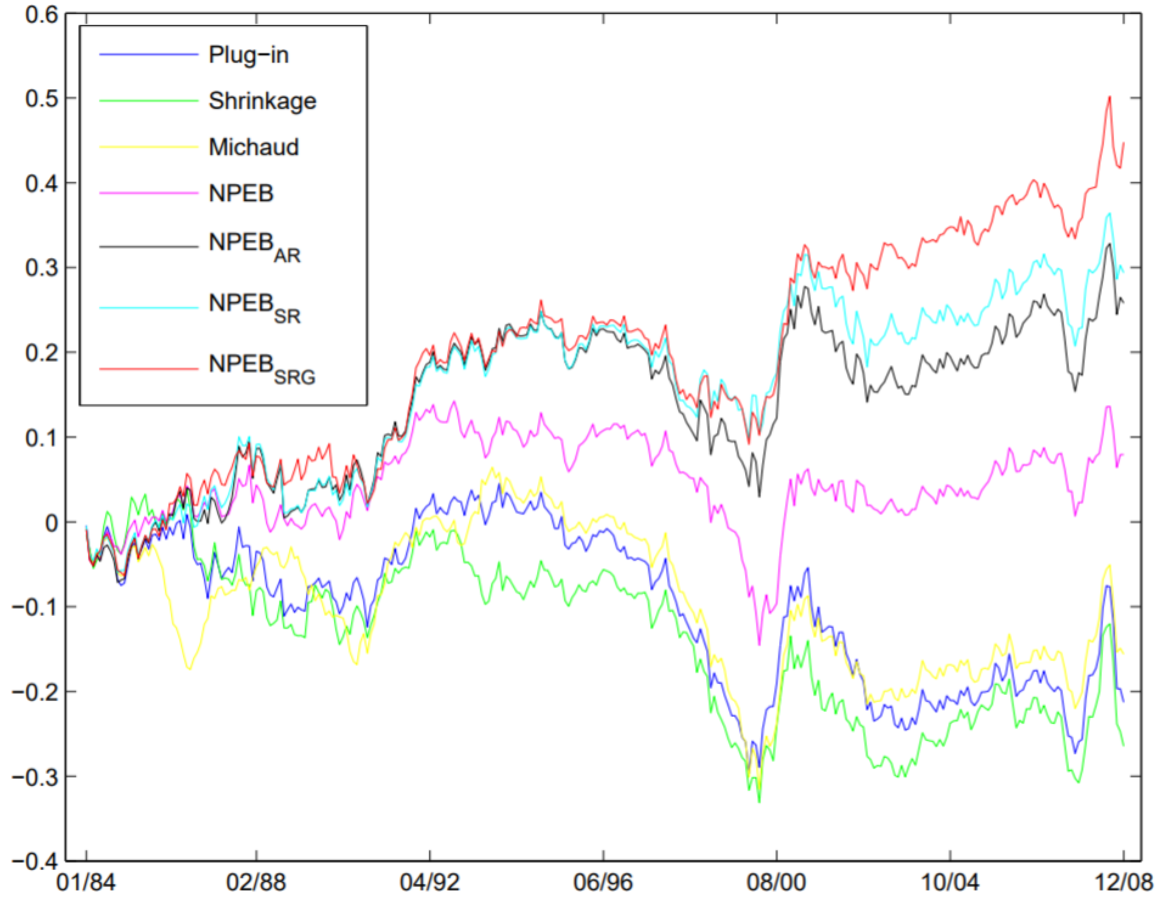
- 把估计的不确定性加入到模型予以考量。
  - $\max_{\omega}\{\mathrm{E}(\omega' r_{t+1}) - \lambda\,\mathrm{Var}(\omega' r_{t+1})\}$
- 这是一个随机优化问题。
  - 使用拔靴法（bootstrapping）来从历史收益率的估计μ_n and V_n。
    - 使用替代指标重新取样
    - 将AR作为模型的收益率
    - 将SR作为模型的收益率
    - 将SR+GARCH作为模型的收益率
- NM:
  - http://numericalmethod.com/blog/2013/02/16/mean-variance-portfolio-optimization-when-means-and-covariances-are-unknown/
  - http://www.numericalmethod.com/javadoc/suanshu/com/numericalmethod/suanshu/model/lai2010/package-summary.html
  - http://redmine.numericalmethod.com/projects/public/repository/svn-algoquant/show/core/src/main/java/com/numericalmethod/algoquant/model/portfoliooptimization/lai2010

未知均值和未知方差



(σ, μ) *curves of different portfolios.*

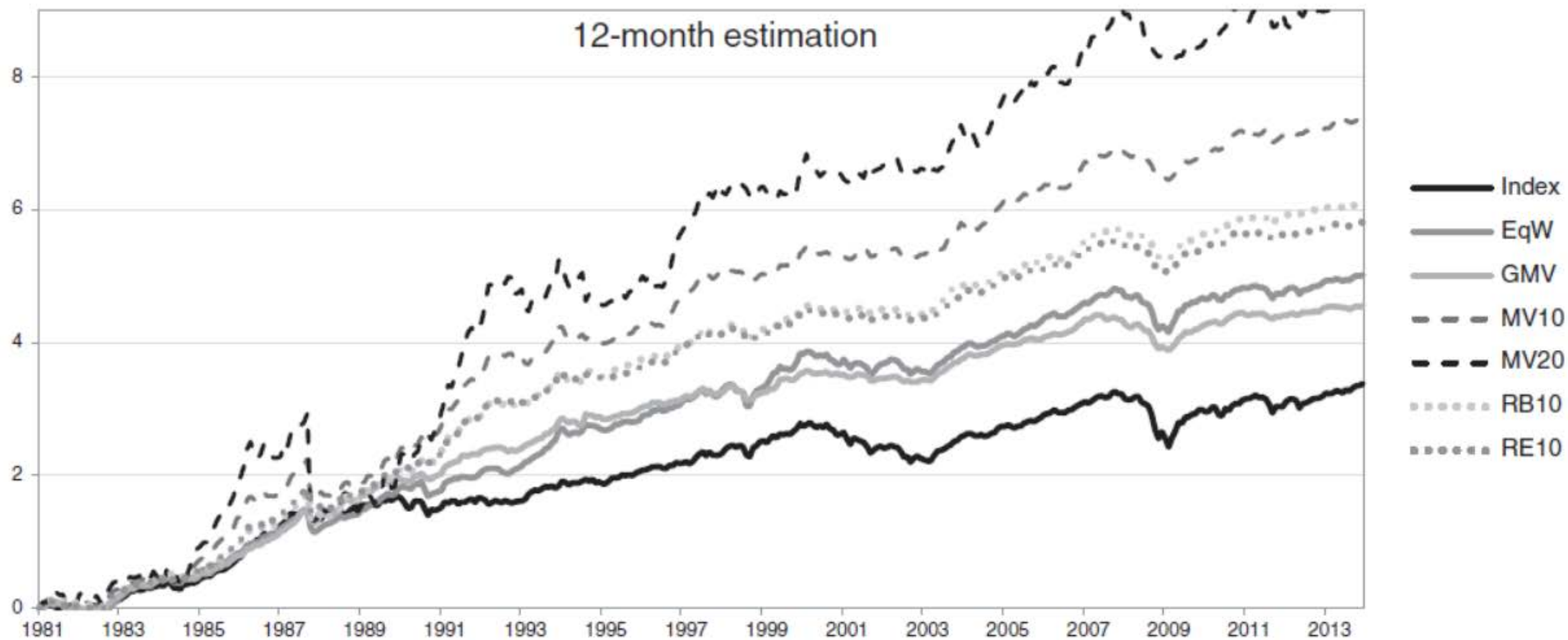
已实现的累计回报—未知均值和未知方差



优化稳健性—在区间内预估错误

- 我们假设输入值、均值和协方差都有内在的不确定性。虽然模型中参数的真实值是确定不知道的，但是假设界是已知的。
- 当考虑不确定集的所有可能性时，最优解代表了最佳的选择。
- 期望收益不确定的稳健公式。
- 通过预期收益中的不确定性来改进公式。
  - $\min_{\omega} \max_{\hat{\mu} \in U} \{\omega' \Sigma \omega - \lambda \hat{\mu}' \omega\}$
  - 它表示最小化所有可能的期望收益值中最差的风险。
- 均值-方差优化问题的改进公式。
  - $\max_{\omega} \min_{(\hat{\mu},\hat{\Sigma}) \in U_{(\hat{\mu},\hat{\Sigma})}} \{\hat{\mu}' \omega - \lambda \omega' \hat{\Sigma} \omega\}$
  - 它表示最大化所有可能的均值和协方差中最差的风险调整收益。

稳健最优化---表现



12-month estimation
0
2
4
6
8
1981
1983
1985
1987
1989
1991
1993
1995
1997
1999
2001
2003
2005
2007
2009
2011
2013
Index
EqW
GMV
MV10
MV20
RB10
RE10

多段的投资组合最优化

- 我们可以在t时（ $t=1,\ldots,T-1$ ）周期性地平衡投资组合。
- 我们的目标函数应该是关于到期时间T的。
  - $\max E[U(W_T)]$
- 在$t=1$阶段, 我们可以通过指定权重来重新平衡投资组合。
- 在 $t=2$ 阶段, 我们知道上一周期的实现收益，因此我们可以用这个信息来重新平衡投资组合。所以，第二阶段的权重是上一阶段（随机）实现的函数。
- 解决方案： 随机规划（stochastic programming）, 动态规划（dynamic programming）

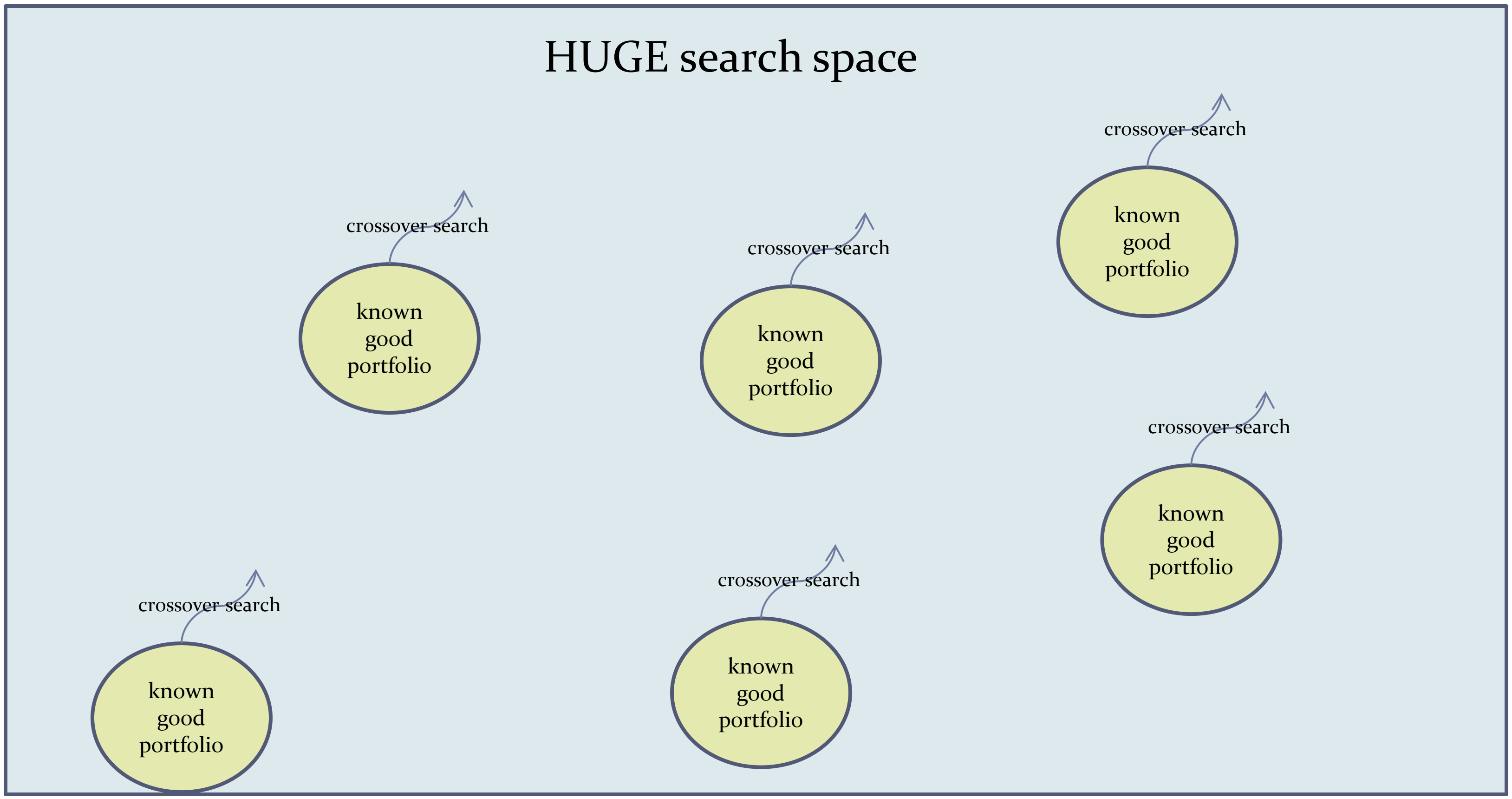
人工智能编程（1）

- 没有参数、没有分析、没有算法预估
- 网格搜索，不需要数学
- 比如说一篮子股票，10个层次，3000只股票，搜索空间为= 10^3000

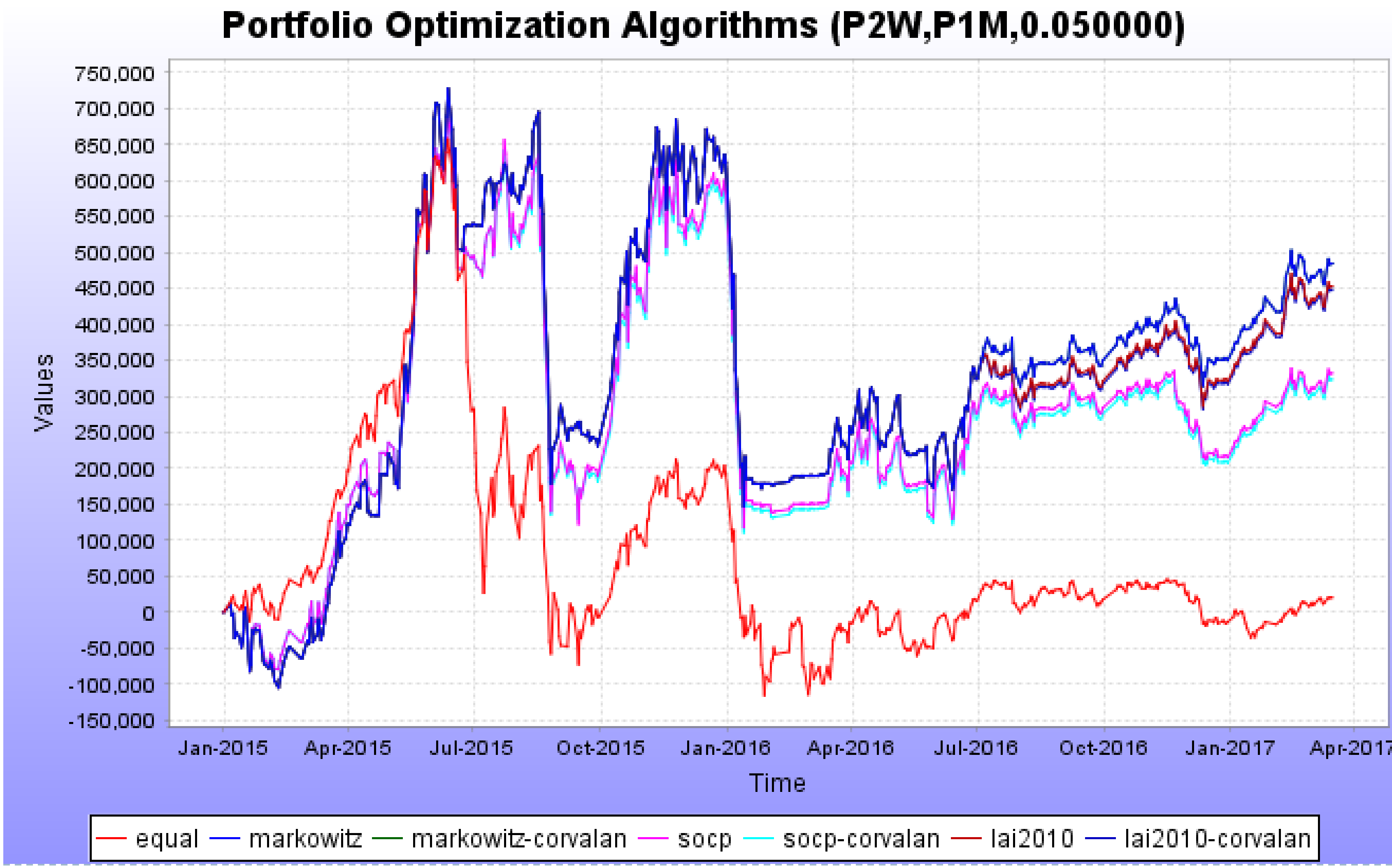
w_{SPY}	w_{TLT}	return r	volat v
0%	100%	-0.7%	8.8%
10%	90%	4.5%	7.6%
20%	80%	9.7%	6.7%
30%	70%	14.9%	6.0%
40%	60%	20.0%	5.7%
50%	50%	25.1%	5.9%
60%	40%	30.3%	6.5%
70%	30%	35.3%	7.5%
80%	20%	40.4%	8.6%
90%	10%	45.5%	9.9%
100%	0%	50.5%	11.3%

人工智能编程（2）

- 用人工智能来提高已知的投资组合收益率



优化算法间的比较


Portfolio Optimization Algorithms (P2W,P1M,0.050000)
Values
750,000
700,000
650,000
600,000
550,000
500,000
450,000
400,000
350,000
300,000
250,000
200,000
150,000
100,000
50,000
0
-50,000
-100,000
-150,000
Jan-2015
Apr-2015
Jul-2015
Oct-2015
Jan-2016
Apr-2016
Jul-2016
Oct-2016
Jan-2017
Apr-2017
Time
equal
markowitz
markowitz-corvalan
socp
socp-corvalan
lai2010
lai2010-corvalan

夏普比率，风险测量的好办法

更好的风险度量方法

- 方差和夏普比率并不是好的风险度量指标。
  - 夏普比率并不区分盈利和亏损的交易，尤其是忽略了它们的概率。
  - 除了均值和方差外，夏普比率没有考虑收益率分布中所有更高的阶矩。
- 其他的风险度量方法:
  - Sortino ratio, $S = \frac{R - T}{DR}$
  - Calmar ratio, $C = \frac{\bar{r}_{36}}{MD}$

夏普的选择

- A和B有同样的均值
- A有更小的方差
- 在均值相同的情况下，夏普会选择方差最小的资产组合
- 因此，在A与B之间，夏普会选择A

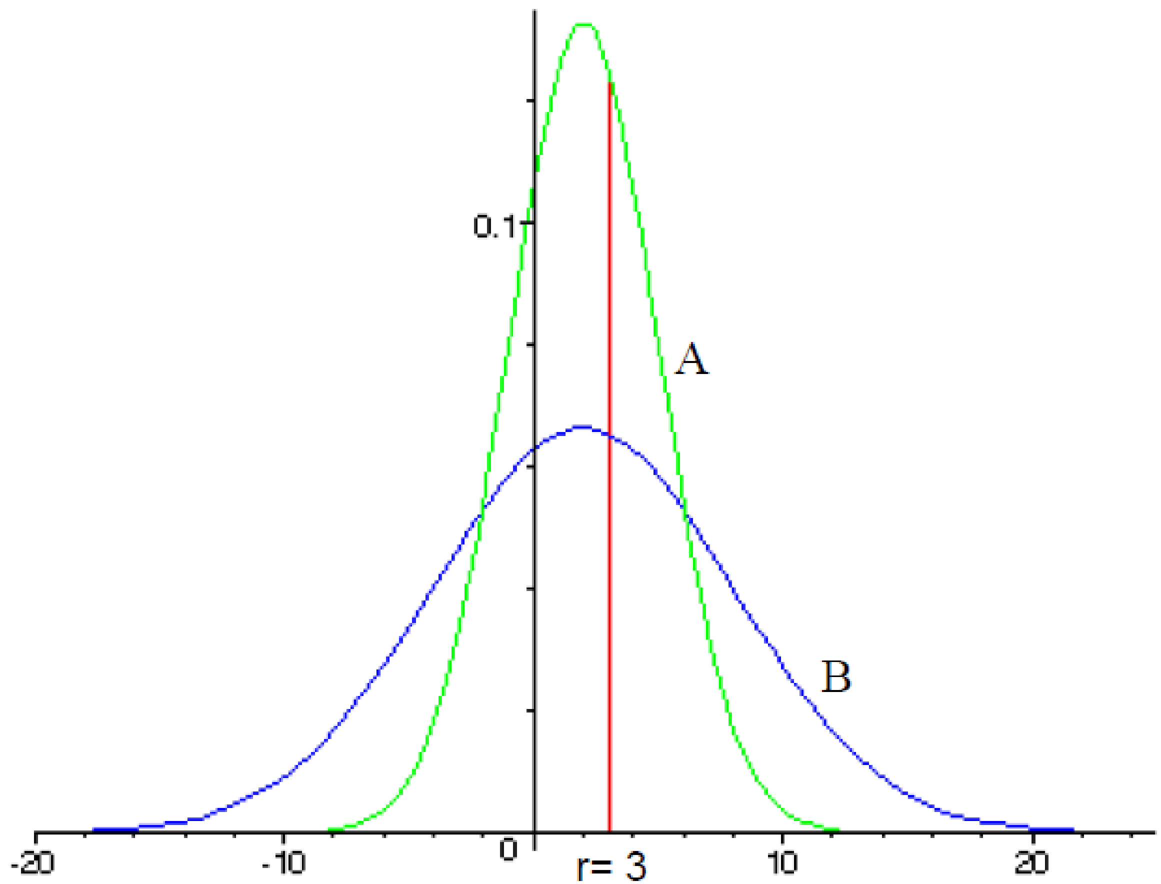
关于避免极端上升和下跌趋势

- 夏普选择协方差最小的资产投资组合来避免极端情况下的损失。
- 然而，对于一个收益回报而言，极端行情带来收益如同带来亏损一样正常。
- 忽略了极端情况的下跌行情将不可避免地忽视了极端情况的上升行情。

潜在收益

- 假设我们根据潜在收益去给投资组合分级，分为A、B级。B级潜在收益高于A级，我们会选择B级。
- 我们会选择有最大协方差的资产组合吗？
- 这就要看直觉了。

举例1：A或B



举例 1: L = 3

- 假设亏损阈值（ Loss Threshold ）= 3
- 图中B在3右侧的部分要多于A在3右侧的部分
  - B: 43%; A: 37%.
- 比较盈利和亏损的可能性
  - B: 0.77; A: 0.59.
- 相比于A，我们更偏向于B

举例1: L = 1

- 假设亏损阈值（ Loss Threshold ） = 1.
- A在1右侧的部分要多于B在1右侧的部分
- 比较盈利和亏损的可能性
  - A: 1.71; B: 1.31.
- 相比于B，我们更偏向于A

业绩衡量要求

- 考虑盈利和损失的概率
- 考虑盈利和损失的大小
- 考虑一个收益率分布的（全部）阶矩

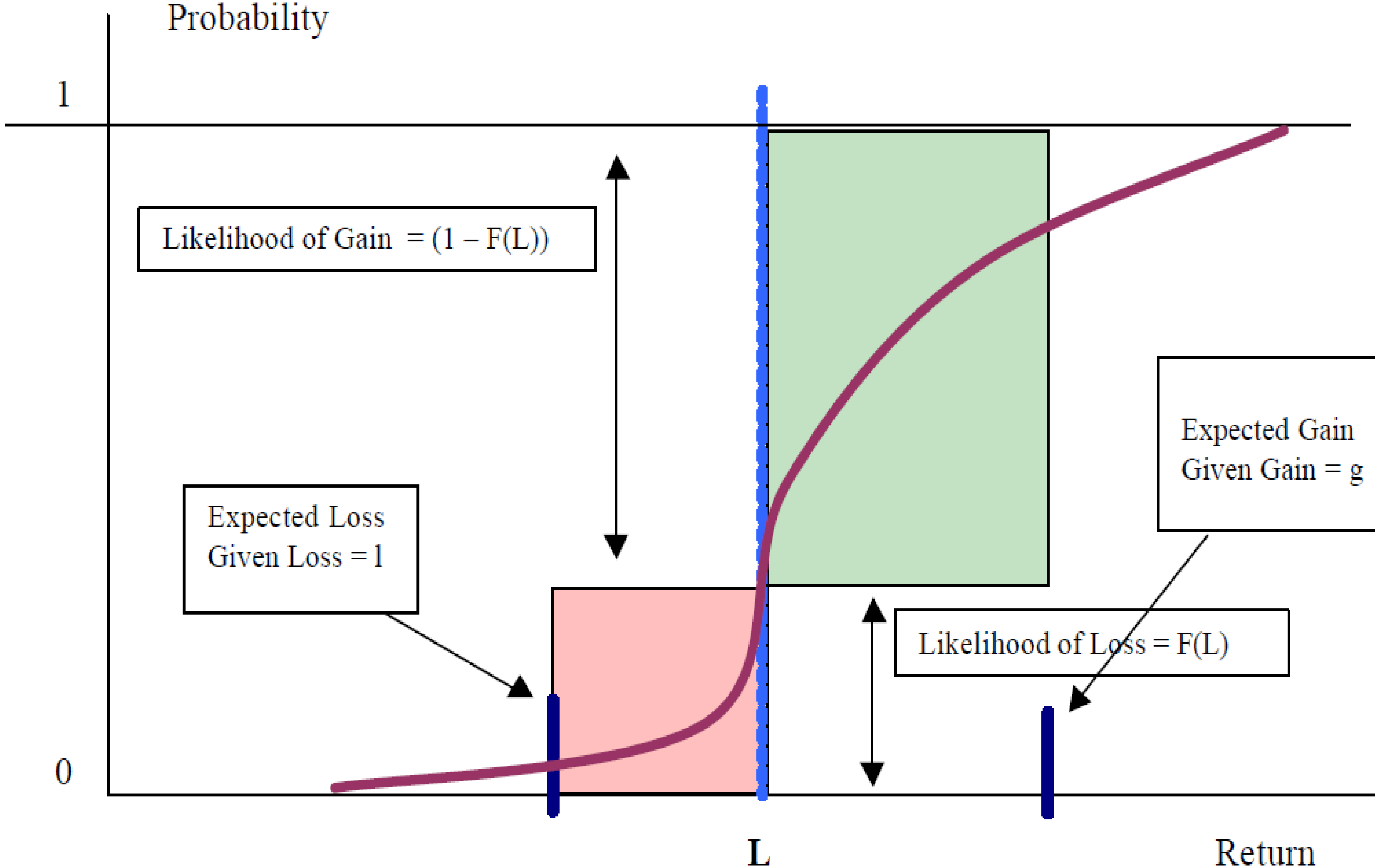
Loss Threshold

- 显然，盈利和亏损概念取决于我们如何定义损失
- 假设 L =亏损阈值（ Loss Threshold ），
  - 当 收益率 < L 时, 视其为损失
  - 当 收益率 > L 时, 视其为盈利

一个尝试

- 加入以下因素进行考虑
  - 盈利和损失概率
  - 盈利和损失大小
- 我们考虑：
  - $\Omega = \frac{E(r|r>L)\times P(r>L)}{E(r|r\leq L)\times P(r\leq L)}$
- $\Omega = \frac{E(r|r>L)\big(1-F(L)\big)}{E(r|r\leq L)F(L)}$

第一个尝试

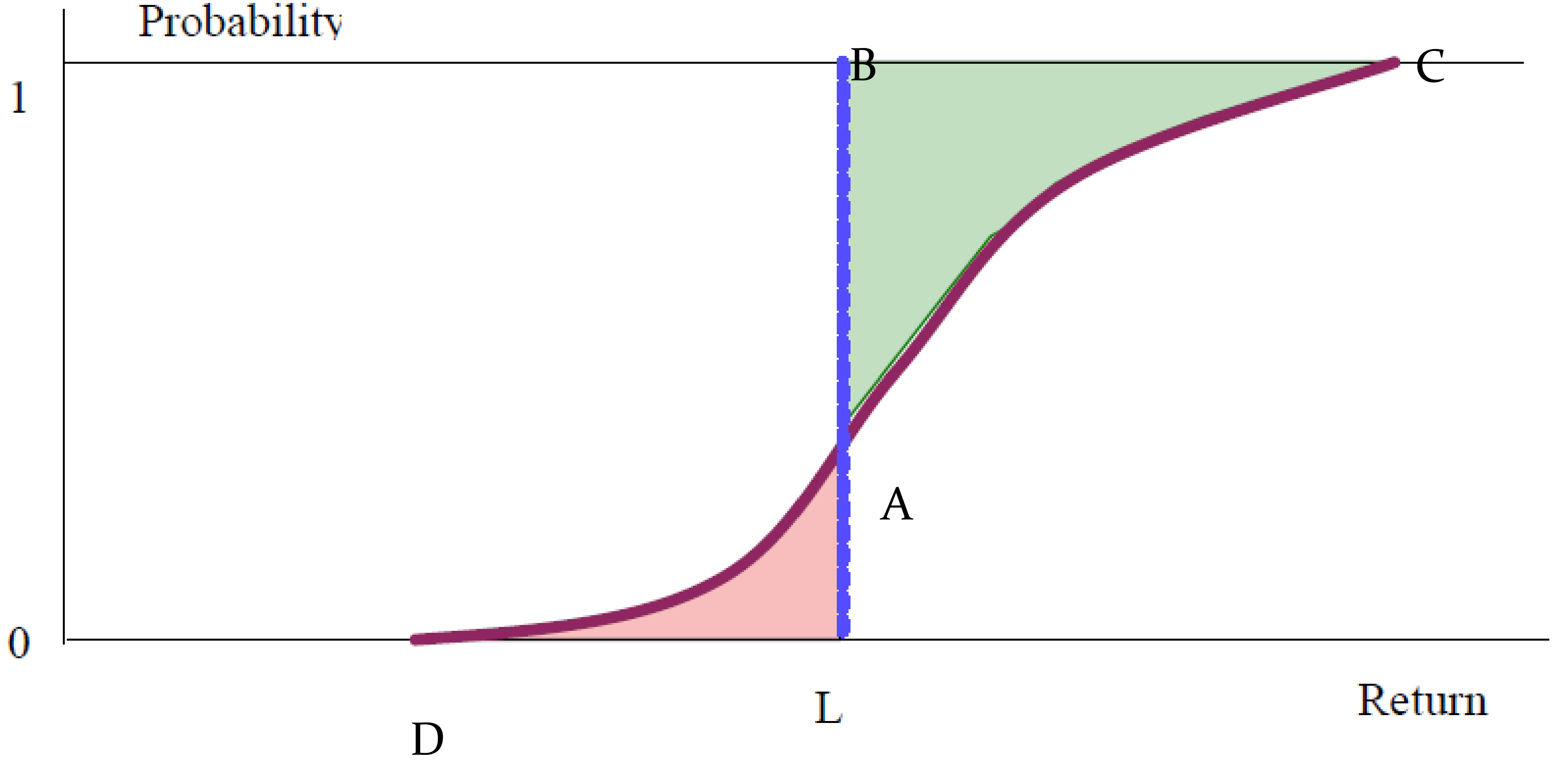


Probability
1
0
Likelihood of Gain = (1 – F(L))
Expected Loss
Given Loss = l
Expected Gain
Given Gain = g
Likelihood of Loss = F(L)
L
Return

第一个尝试不充分

- 为什么是 F(L)?
- 没有使用整体分布的信息
  - 因此忽略了高阶矩

另一个尝试



Omega 的概念

- Ω引入了极限的概念
- Ω使用整体分布
- Ω的定义:
  - $\Omega = \frac{ABC}{ALD}$
- $\Omega = \frac{\int_{L}^{b=\max\{r\}}[1-F(r)]dr}{\int_{a=\min\{r\}}^{L}F(r)dr}$

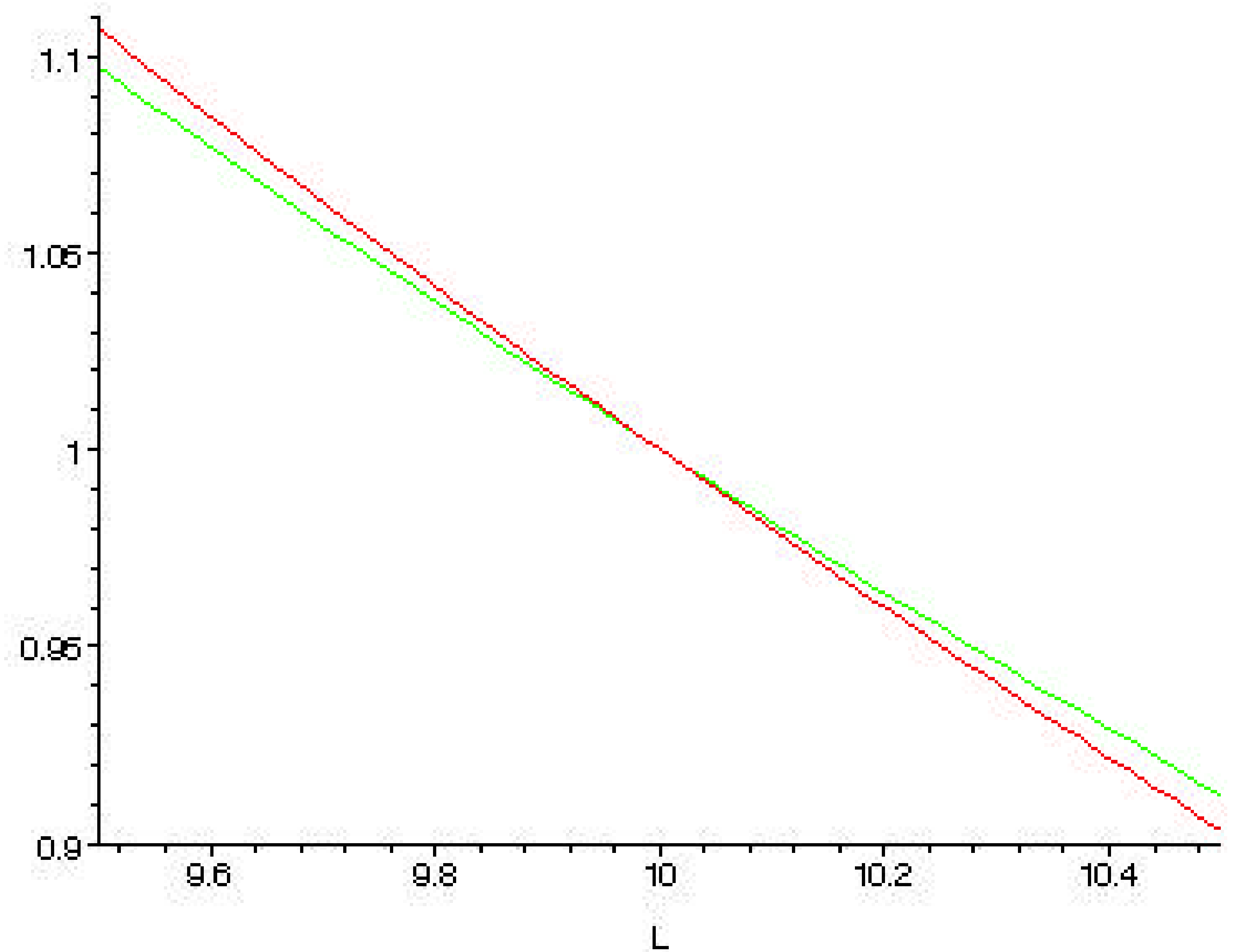
Intuition

- Omega 是概率加权的收益大小与概率加权损失大小的比值
- Omega 考虑了盈利交易和损失交易大小和概率
- Omega 考虑了所有阶矩因为它的定义包括了整体分布

Omega 的优势

- 没有参数(estimation)
- 不需要预测高阶矩
- 适用于所有分布
- 使用亏损阈值（ Loss Threshold ）函数来衡量表现而非像夏普比率那样仅仅使用一个数来衡量
- 和收益率分布一样平滑
- 是单调递减的

Omega 举例



另一方面看 Omega

- $$\Omega = \frac{\int_{L}^{b=\max\{r\}}[1-F(r)]dr}{\int_{a=\min\{r\}}^{L}F(r)dr}$$
- $$= \frac{E[\max(x-L,0)]}{E[\max(L-x,0)]}$$
- $$= \frac{e^{-r_f}E[\max(x-L,0)]}{e^{-r_f}E[\max(L-x,0)]}$$
- $$= \frac{C(L)}{P(L)}$$

Options Intuition

- 分子: 收益高于L时，取得收益的成本
- 分母: 收益低于L时，保护收益的成本
- 风险度量: 使用卖方期权价格作为保护成本比使用方差更为普遍。

如何做得更好？

- Sharpe Ratio中的超额收益比Omega中的$C(L)$更加直观。
- Omega中使用卖方期权价格来衡量风险的方式要优Sharpe Ratio中使用方差来衡量风险的方式。

Sharpe-Omega

- $\Omega_S = \frac{\bar{r} - L}{P(L)}$
- 在这个定义中，我们结合了夏普比率（ Sharpe Ratio ）和 Omega的优势。
  - 超额收益的界定是清晰的
  - 风险也被更好的衡量
- Sharpe-Omega 更加直观。
- Ω_S 和 Ω 对资产组合的分类方法一致。

Sharpe-Omega and 阶矩（ Moments ）

- 需要注意的是分子只涉及收益率分布的一阶矩（平均值）。
- 分母考虑了整个分布，包括方差和所有高阶矩。

超越均值方差优化

非线性、非凸投资组合优化

- 一般来说，夏普最优投资组合不同于Omega最优投资组合。

Omega优化

- $$\begin{cases} \max\limits_{x} \Omega_S(x) \\ \sum_i^n x_i E(r_i) \geq \rho \\ \sum_i^n x_i = 1 \\ {x_i}^l \leq x_i \leq 1 \end{cases}$$
- 最低持有: $x^l = \left({x_1}^l, \dots, {x_n}^l\right)'$

最优化方法

- 非线性规划（ Nonlinear Programming ）
  - 惩罚法（ Penalty Method ）
- 全局优化（ Global Optimization ）
  - 禁忌搜索（ Tabu search ） (Glover 2005)
  - Threshold Accepting algorithm (Avouyi-Dovi et al.)
  - MCS algorithm (Huyer and Neumaier 1999)
  - 模拟退火算法（ Simulated Annealing ）
  - 遗传算法（ Genetic Algorithm ）
- 整数规划（ Integer Programming ） (Mausser et al.)

三资产举例

- $x_1 + x_2 + x_3 = 1$
- $R_i = x_1 r_{1i} + x_2 r_{2i} + x_3 r_{3i}$
- $= x_1 r_{1i} + x_2 r_{2i} + (1 - x_1 - x_2) r_{3i}$

罚函数方法（ Penalty Method ）

- $F(x_1, x_2) = -\Omega(R_i) + \rho\{[\min(0, x_1)]^2 + [\min(0, x_2)]^2 + [\min(0,1 - x_1 - x_2)]^2\}$
- 可以运用Nelder-Mead, 一个需要初始猜测的单纯形（ Simplex ）算法。
- F不需要是可微的。
- 可以进行随机重启来搜索全局最优。

Threshold Accepting Algorithm

- 一种局部搜索算法。
  - 围绕当前最佳解决方案探索潜在的候选方案。
- 通过选择低于当前最优的解来“避开”局部极小值。
  - 这是与爬山算法（ hilling climbing algorithm ）非常鲜明的对比。

Objective

- 目标函数
  - $h: X \rightarrow R, X \in R^n$
- 最优化
  - $h_{\text{opt}} = \max_{x \in X} h(x)$

初始（ Initialization ）

- 初始化 n (迭代次数) 和 步骤（$step$）.
- 初始化阈值序列$th_k, k = 1, \ldots, step$
- 起始点: $x_0 \in X$

阈值（Thresholds）

- 模拟一个资产组合
- 计算资产组合间的距离
- 将距离从小到大进行排序
- 选择他们的第一阶数字（first step number）作为阈值

Research

- $x_{i+1} \in N_{x_i}$ (x_i的邻值)
- 阈值: $\Delta h = h(x_{i+1}) - h(x_i)$
- : If $\Delta h > -th_k$ set $x_{i+1} = x_i$
- 持续接受直到我们完成了最后（最小的）阈值。
  - $h(x_i) \approx h_{opt}$
- 通过蒙特·卡罗方法来评估h.

人工智能（AI）– 遗传规划（Genetic Programming）

- 传统方法难以对任意的、非凸的、不可微的、非连续的、噪声的目标函数进行优化。我们寄希望于人工智能、启发法（ heuristics ）和模拟。
- 在遗传算法中，一个优化问题的候选解群（也称为个体、生物或显性性状）逐渐进化成更好的解决方案。每个候选解有一组属性（染色体或基因型），而这些属性可以突变和修改；通常，这些解是通过一串0与1的二进制来表现的，但其他编码也是可能的。
- NM的遗传编程框架:
  - http://www.numericalmethod.com/javadoc/suanshu/com/numericalmethod/suanshu/optimization/multivariate/geneticalgorithm/package-summary.html

差分进化算法（ Differential Evolution ）

- 差分进化算法运用于多维实值函数，但是不使用优化问题的梯度，这意味着差分进化算法不需要像诸如梯度下降（gradient descent）和拟牛顿法(quasi-newton)的经典优化方法那样，要求优化问题是可微的。因此，差分进化算法也可以运用于解决一些优化问题，比如非连续性、噪声的、随时间变化等。
- 差分进化算法通过保持候选解群，并根据其简单公式组合已有的候选解来创建新的候选解，然后保留在此优化问题中具有最高评分或最为合适的候选解，进而优化该问题。这样，优化问题就被视为一个只提供衡量候选解决方案质量的黑盒，而并不需要梯度。
- NM:
  - http://numericalmethod.com/blog/2011/05/31/strategy-optimization/
  - http://www.numericalmethod.com/javadoc/suanshu/com/numericalmethod/suanshu/optimization/multivariate/geneticalgorithm/minimizer/deoptim/DEOptim.html