

Momentum과 Markowitz: 황금 조합

Wouter J. Keller, Adam Butler, Ilya Kipnis, 2015년

5월 16일(6월 4일 수정: v0.99d)

¹

추상적인

Markowitz(1952)가 도입한 평균 분산 최적화(MVO)는 종종 우아하지만 비현실적인 이론으로 제시됩니다. MVO는 "불안정하고 오류를 최대화하는" 절차(Michaud 1989)이며 "거의 항상 단순한 1/N 포트폴리오에 의해 패배합니다"(DeMiguel, 2007). 그리고 Ang(2014)을 인용하자면: "평균 분산 가중치는 끔찍하게 수행됩니다. ... 최적의 평균-분산 포트폴리오는 추정된 평균, 변동성 및 자산 수익률의 상관 관계의 복잡한 함수입니다. 추정해야 할 많은 매개변수가 있습니다. 최적화된 평균 분산 포트폴리오는 이러한 입력에 작은 오류가 있을 때 폭발할 수 있습니다. ...".

우리의 의견으로는 MVO는 훌륭한 개념이지만 이전 연구는 공매도를 허용하고 잘못된 지정된 추정 범위를 적용했기 때문에 실패할 운명이었습니다. 예를 들어 Ang은 평균과 분산을 추정하기 위해 60개월의 형성 기간을 사용했으며 Asness(2012)는 과거 최고의 자산이 종종 미래의 최악의 자산이 되는 이 시간 척도에서 가격이 평균 회귀한다는 것을 분명히 보여주었습니다.

이 백서에서는 모멘텀 팩터를 가장 잘 수확하기 위해 짧은 록백 기간(최대 12개월)을 적용하여 MVO 매개변수를 추정합니다. 또한 최적화를 안정화하기 위해 롱 전용 포트폴리오 가중치와 같은 상식적인 제약을 도입할 것입니다. 또한 자산 수가 록백 기간 수보다 훨씬 많은 경우를 처리하기 위해 R로 프로그래밍된 Markowitz의 CLA(Critical Line Algorithm)의 공개 구현을 소개합니다.

우리는 우리의 모멘텀 기반 장기 전용 MVO 모델을 고전적 자산 배분 (CAA)이라고 부르고 100년 동안의 데이터(1915년 1월-2014년 12월) 동안 다양한 글로벌 다중 자산 유니버스를 사용하여 간단한 1/N 동일 가중 포트폴리오와 성능을 비교합니다. . 결말을 망칠 위험을 무릅쓰고 우리는 CAA가 항상 단순한 1/N 모델을 큰 차이로 이겼다는 것을 보여줍니다.

1. 소개

Markowitz와 Lehman Brothers에 대해 무지한 고등학생 Harry가 "인형을 위한 금융"이라는 수업에 등록했다고 상상해 보십시오. 2008년 8월 말에 그의 교사는 소위 Efficient Frontier(수익률 대 변동성)를 계산하라고 요청했습니다.) SPY(S&P500 ETF)와 TLT(장기 Treasury ETF)를 혼합하여 목표 변동성(TV)이 10%인 최고 수익률 포트폴리오를 결정합니다.

Harry는 두 자산에 대한 월간(총) 수익률 데이터만 사용하고 직전 4개월 동안 최적의 포트폴리오를 조사하기로 선택합니다. 그는 SPY 및 TLT에 대한 4개월 수익(2008년 5월 - 8월)만 있는 간단한 스프레드시트로 시작합니다(그림 참조). 1.

¹ Lennart Appels, Victor DeMiguel, Frank Grossman, Winfried Hallerbach, Jan Willem Keuning, Clarence Kwan, Steve LeCompte, Hugo van Putten에게 영감과 의견을 주신 데 감사드립니다. 모든 오류는 우리의 것입니다.

보고	스파이	TLT
2008-05-30	1.5%	-2.7%
2008-06-30	-8.3%	2.7%
2008-07-31	-0.9%	-0.4%
2008-08-29	1.5%	2.7%

그림 1 SPY 및 TLT의 월별 수익률(TR, EOM)

각 가중치 조합 ($w_{SPY}=0, 10, \dots, 90$, SPY의 경우 100% 및 w_{TLT} 의 보완)에 대해 Harry는 월별 포트폴리오 수익을 구성하고 해당 평균 포트폴리오 수익을 관찰했습니다 ($r_P = w_{SPY} \cdot r_{SPY} + w_{TLT} \cdot r_{TLT}$) 및 간단한 스프레드시트 기능을 사용하여 4개월 동안 r_P 의 변동성.

그림을 참조하십시오. 2. 그런 다음 그는 EF(Efficient Frontier)라는 11가지 가중치 조합 모두에 대한 평균 수익률과 변동성을 모두 표시했습니다. 그림을 참조하십시오. 삼.

wSPY	wTLT	r을 반환	변동성 v 9.1%
0%	100%	6.7%	
10%	90%	4.2%	7.5%
20%	80%	1.7%	6.3%
30%	70%	-0.8%	5.8%
40%	60%	-3.4%	6.1%
50%	50%	-6.0%	7.1%
60%	40%	-8.6%	8.5%
70%	30%	-11.3%	10.2%
80%	20%	-14.1%	12.1%
90%	10%	-16.8%	14.1%
100%	0%	-19.6%	16.2%

그림 2 SPY/TLT 가중치가 다양한 포트폴리오의 수익률 및 변동성(2008년 5월-8월)

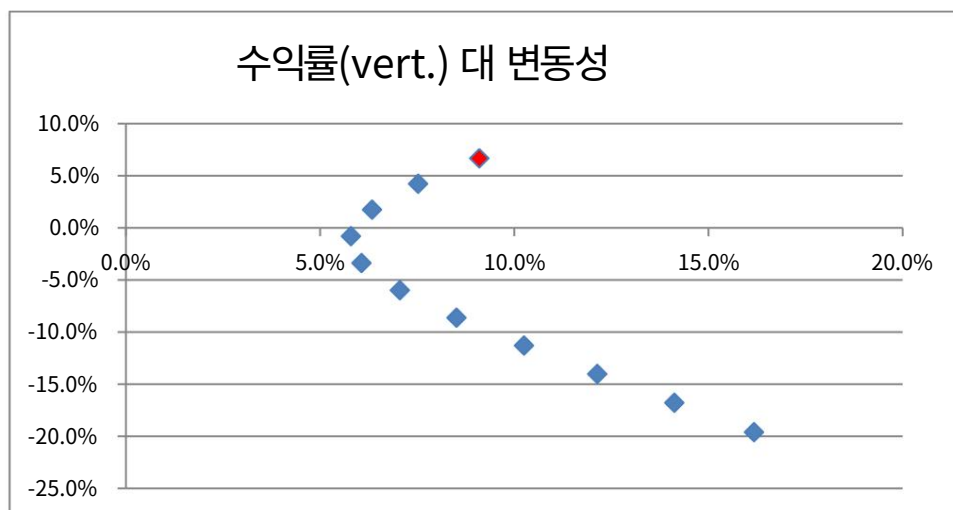


그림 3 2008년 8월 29일의 효율적 프론티어

Harry는 목표 변동성 10%를 목표로 하고 있으므로 향후 포트폴리오에 대해 EF의 맨 오른쪽 상단(빨간색 점)으로 표시되는 가중치 혼합(평균 수익률 6.7% 및 변동성 9.1%)을 선택합니다. 이 점은 $wSPY=0\%$ 및 $wTLT=100\%$ 에 해당합니다. 9월 말, Harry가 최적의 포트폴리오 성과를 검토했을 때 그는 포트폴리오가 긍정적인 수익을 창출했으며 그 달에 대부분의 자산 클래스가 경험한 대규모 충동에 대한 탄력성을 입증했음을 관찰했습니다. 사실, 그가 다음 달 말에 이 프로세스를 계속 따랐다면 그는 자신의 포트폴리오에 TLT만 있는 2008년 총돌의 '묘지'를 지나서 휘파람을 불었을 것입니다.

Harry의 원래 연습에서 2009년 8월 말까지 12개월을 빨리 감아 보겠습니다. Harry가 4개월(2009년 5월-8월)의 동일한 록백으로 동일한 분석을 실행했을 때 그는 다음과 같은 수익률과 변동성을 관찰했습니다(그림 4 및 5 참조).

스파이	wTLT	r을 반환	휘발성 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%

그림 4 SPY/TLT 가중치가 다양한 포트폴리오의 수익률 및 변동성(2009년 5월-8월)

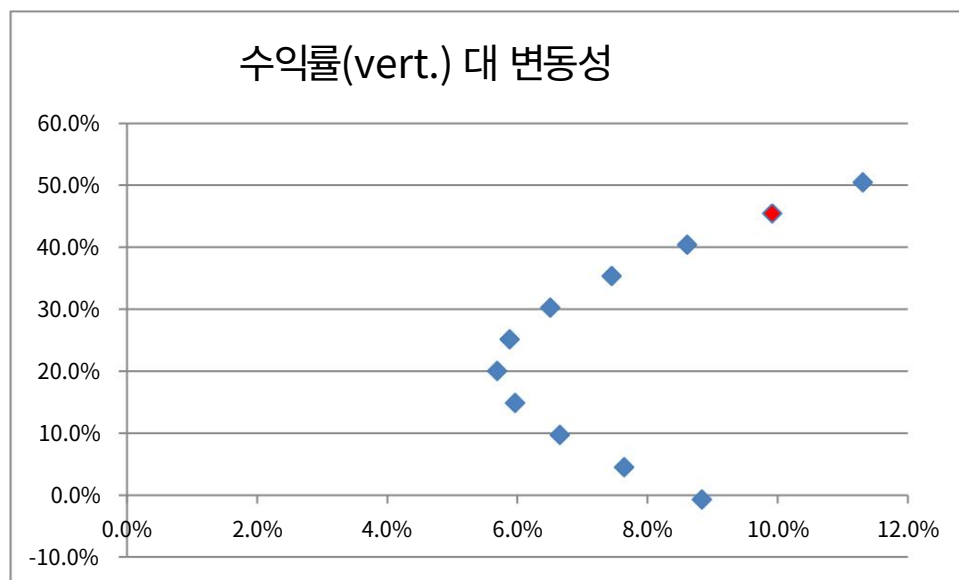


그림 5 2009년 8월 31일의 효율적 프론티어

이제 목표 변동성이 10%인 Harry는 90% SPY와 10% TLT(그림 5의 빨간색 점) 포트폴리오를 선택합니다. 이것은 Harry의 포트폴리오가 주식 회복의 혜택을 받았기 때문에 잘 작동했습니다.

Harry의 선택에서 관찰된 긍정적인 결과는 잠재적인 기회를 암시합니다. 하지만 해리는 무엇을 했습니까? 실제로 Harry는 Markowitz의 MVO(Mean Variance Optimization)를 사용했는데 나중에 표시하겠지만 몇 가지 제약이 있습니다. 구체적으로 그는 양의 가중치(즉, 공매도 없음)가 있는 포트폴리오만 고려하고 매개변수 추정을 위해 짧은 록백 기간을 사용합니다. (그는 또한 이산 가중치를 부과하지만 이것은 다른 논문의 주제입니다).

그러나 잠시만요, MVO가 "불안정하고 오류를 최대화하는" 절차(Michaud 1989)가 아니었습니까? 거의 항상 간단한 $1/N$ 포트폴리오(DeMiguel, 2007), 특히 제한된 데이터(여기서는 4개월)에 의해 압도당했습니다. 인용 Ang(2014): "평균-분산 가중치는 끔찍하게 수행됩니다... 최적의 평균 분산 포트폴리오는 추정된 평균, 변동성 및 자산 수익의 상관 관계의 복잡한 함수입니다. 추정해야 할 많은 매개변수가 있습니다. 최적화된 평균 분산 포트폴리오는 이러한 입력에 작은 오류가 있을 때 폭발할 수 있습니다."

“

위의 진술을 간단한 예와 어떻게 조사시킬 수 있습니까? 요약하면 다음과 같습니다. MVO는 훌륭한 개념이지만 이전 연구는 단기 판매를 허용하고 잘못 지정된 추정 범위를 적용했기 때문에 실패할 운명이었습니다. 예를 들어 Ang은 평균과 분산을 추정하기 위해 60개월 형성 기간을 사용했으며 Asness(2012)는 가격이 과거 최고의 자산이 미래에는 종종 최악의 자산이 되는 이 시간 척도에서 회귀를 의미한다는 것을 명확하게 보여주었습니다. 이 경우 DeMiguel(2007), Kritzman(2010), Jacobs(2013) 및 Ang(2014)이 보여준 것처럼 MVO가 단순한 $1/N$ 투자에 비해 저조한 것은 놀라운 일이 아닙니다. 또한 Ma(2002)는 MVO의 불안정성의 대부분이 우리 자신을 long-only로 제한할 때 사라짐을 보여줍니다.

롤링 추정 기간의 사용은 롤링 추정이 후속 포트폴리오 보유 기간 동안 지속되어야 함을 의미하므로 역사적으로 수익률 지속성과 관련된 롤링 추정 기간의 사용을 옹호합니다. 물론 수익률 지속 현상은 모멘텀 팩터(예: Faber 2007, Hurst, 2012, Butler 2012, Antonacci 2011 및 Asness 2014 참조)와 가장 일치하며 최적의 포트폴리오 형성 기간은 1~12개월입니다. 변동성과 상관 관계의 지속성에 대해서도 마찬가지로(Keller 2012는 "일반화된 모멘텀"이라고 함).

다행스럽게도 해리가 보여준 것처럼 자산 변동성 상관 관계를 계산할 필요조차 없습니다. 결과 포트폴리오(그림 2 참조)의 (r, v) 를 계산하고 효율적인 프론티어에 도달하기 위해 포트폴리오 자산(그림 1 참조)의 이산 조합을 통해 순열하여 평균 분산 최적 포트폴리오 가중치를 근사화할 수 있습니다(그림 2 참조). 그림 3). 추정 기간이 1-12개월 범위에 있는 한 모멘텀은 수익률 추정치를 알려줄 것입니다. 반대 수익률 패턴(즉, 음의 상관관계)을 가진 자산을 결합하면 변동성이 더 낮은 포트폴리오(즉, v 가 더 낮은 EF 포인트)가 되기 때문에 변동성과 상관관계도 마찬가지로입니다.

Harry가 간단한 스프레드시트로 설명했듯이 "단일 행렬을 사용한 복잡한 2차 최적화" 등에 대해 신경 쓸 필요가 없습니다. 사실 위의 예제와 같이 Markowitz 모델은 매우 간단합니다. Harry가 했던 것처럼 지난 몇 년간의 자산 수익률부터 시작하여 다양한 자산 가중치에 대한 포트폴리오 (r, v) 포인트를 계산하여 효율적인 프론티어를 간단히 계산합니다.

몇 달.

2. 효율적인 프론티어 계산: 직접 및 간접 접근

평균 분산 최적화는 Markowitz의 현대 포트폴리오 이론(MPT, Markowitz, 1952 참조)의 핵심입니다. 위의 예에서 설명한 것처럼 효율적 프론티어(EF)의 개념을 기반으로 합니다. N 개의 자산으로 구성된 대규모 유니버스에서도 이론상 가능한 모든 롱 전용 가중치 조합을 조사하고 특정 짧은 록백 기간(최대 12개월) 동안 포트폴리오의 평균 수익률(r)과 변동성(v)을 계산할 수 있습니다. EF는 단순히 이러한 모든 (r, v) 지점을 포함하는 봉투입니다. 총알의 왼쪽 크기 타입에 최소 변동성(MV) 솔루션이 있는 위에서 볼 수 있는 전형적인 총알 모양을 가짐을 알 수 있습니다.

EF 글머리 기호의 하단(MV 지점 아래)에 있는 모든 솔루션은 비슷한 위험에 더 높은 수익을 제공하는 포트폴리오에 의해 지배(더 나은 r , 더 낮은 v)되기 때문에 "비효율적"입니다. 따라서 진정한 "효율적" 프론티어(EF)는 엔벨로프의 상단 부분입니다. 대부분의 "퀀트"는 변동성 단위당 최대 초과 수익을 생성하는 최대 샤프 비율(MSR) 포트폴리오에 중점을 둡니다. 이 MSR 포트폴리오는 r 축에서 무위험 금리를 통과하는 선에 접하는 EF 지점에 해당합니다.

또는 투자자의 효용 함수에 대한 EF 접선의 유사한 지점에 관심이 있을 수 있습니다. 예를 들어 Ang(2014) 참조.

우리는 연간 목표 변동성 (TV) 을 사용하여 EF에서 최적의 지점을 찾고 적절한 가중치 조합을 찾아 일정한 수준의 위험에 집중하는 것을 선호합니다. 따라서 공격적인 투자자는 TV=10%를 선택할 수 있고 보다 보수적인 투자자는 TV=5%를 선택할 수 있습니다. 이는 CAA(Classical Asset Allocation)와 같은 EF 기반 모델의 큰 장점 중 하나입니다. 그러나 우리가 하는 것처럼 목표 변동성을 사용하는 대신 목표 수익률, 샤프 비율 최대화 또는 분산(또는 변동성) 최소화를 위해 동일한 EF(따라서 CAA 모델)를 사용할 수 있습니다. , 더 일반적입니다. 그것은 모두 위험과 수익에 대한 귀하의 선호도에 달려 있습니다.

EF는 어떻게 계산합니까? 말했듯이, 그것은 단순히 모든 가능한 포트폴리오의 (r, v) 쌍의 상위 포락선입니다. 포트폴리오 수익률 r 의 계산은 항상 매우 간단합니다. 다양한 자산 수익률에 포트폴리오 가중치를 적용하면 됩니다. 포트폴리오 변동성 v 를 계산하는 두 가지 방법이 있습니다. 공분산 행렬을 사용하는 간접 행렬입니다. 대부분의 2차 옴티마이저에서 사용하는 후자의 방법은 록백 기간 T 에 대한 자산 공분산 행렬 C 의 계산을 필요로 합니다. 동일한 록백 기간 동안 C 를 추정할 때 C 를 사용하는 간접 방법은 다음과 같다는 것을 쉽게 알 수 있습니다.² Harry가 스프레드시트에서 사용한 직접적인 방법입니다.

N (자산 수)이 T (관찰 수)보다 훨씬 크면 $N \gg T$ 이므로 포트폴리오 변동성에 대해 Harry의 방법을 사용하는 것이 계산상 더 효율적입니다(가중치 세트가 주어짐). $N \ll T$ 일 때 공분산 행렬을 기반으로 변동성을 계산하는 것이 좋습니다. 그만큼

² 행렬 표기법: C 가 N 자산에 대한 과거 $N \times N$ 공분산 행렬인 경우 X 는 T 관측치(지난 12개월을 사용하는 경우 $T=12$)가 있는 수익의 $T \times N$ (demeaned) 행렬이고 w 는 a 의 가중치 N -벡터입니다. $C = X'X/T$ 이므로 해당 포트폴리오의 분산은 $v^2 = w'X'Xw/T = w'Cw$ 입니다. 따라서 가중치 w 가 있는 포트폴리오의 역사적 분산(따라서 변동성 v)은 지난 12개월 동안 포트폴리오의 역사적 분산(즉, Xw 의 분산)을 사용하여 직접 계산하거나 공분산 행렬 C 를 먼저 계산하여 간접적으로 계산할 수 있습니다. 그리고 거기에서 분산 $v^2 = w'Cw/T$. 이것은 또한 분산(및 변동성)이 가중치 w 의 2차 함수($w'Cw$)임을 보여줍니다.

결과는 동일합니다. 월별 모멘텀 모델($T \leq 12$)에서 Harry의 방법은 $N=12$ 보다 큰 모든 유니버스에 대해 더 효율적입니다. 또한 관련된 공분산 행렬이 없기 때문에 이해하기가 훨씬 간단합니다. 마지막 $T \leq 12$ 에 대한 포트폴리오의 역사적 변동성만 있으면 됩니다.

몇 달.

Harry의 방법은 자산의 수 N 이 T 보다 훨씬 큰 경우에도 잘 작동합니다. 단순히 임의의 자산 가중치(장기 전용)를 생성하고 포트폴리오 수익률 r 과 변동성 v 를 예를 들어 마지막 T 관측치에 대해 계산하고 이를 플로팅합니다(r, v) EF 다이어그램에서 생성된 모든 가중치에 대한 쌍.

"추정 평균, 변동성, 자산 수익률의 상관관계의 복잡한 함수"도 없고 "추정할 많은 매개변수"도 없습니다(Ang, 2014).

Harry의 예에서 알 수 있듯이 EF의 계산(직접 방법 사용)은 이보다 더 간단할 수 없습니다.

그리고 우리가 보여줄 것처럼 모멘텀 기반 전술적 접근 방식($T \leq 12$ 개월 룩백)과 장기 전용 제약 조건의 조합은 Markowitz의 MVO를 "불안정하고 오류를 최대화하는 절차"에서 강력한 포트폴리오 형성 방법으로 변환합니다.

또한 주어진 목표 변동성에 대한 EF와 해당 최적 가중치는 수익률 및 변동성의 규모 변화(예: 월간 또는 연간)에 대해 변하지 않는 반면 수익률 수준의 병렬 변화에도 독립적임을 쉽게 보여줄 수 있습니다. 모든 자산에 대해(단순히 EF를 위 또는 아래로 이동)3.

따라서 최적의 가중치(주어진 목표 변동성이 있는 EF의 지점에 해당)는 이해하기 쉽고 특히 장기 전용 포트폴리오의 경우 매우 강력합니다. 이것은 될 수 있다

엑셀로 시연했다. $N=9$ 자산이 있는 유니버스와 $T=4$ 개월의 짧은 룩백이 있더라도

Harry가 사용한 것처럼(따라서 $T \leq N$) Excel에서 수천 개의 무작위(음수가 아닌) 가중치를 생성하여 EF를 결과 산란의 상위 포락선으로 볼 수 있습니다. 4×9 수익이 주어지면 각 가중치 세트에 대한 포트폴리오(r, v)를 계산하십시오. 이제 "최적화된 평균 분산 포트폴리오는 이러한 입력에 작은 오류가 있을 때 폭발할 수 있습니다."

(Ang, 2014). $N \gg T$ 인 경우에도 EF(및 주어진 TV에 대한 최적의 가중치)는 우리의 가정($T \leq 12$ 개월 및 장기 전용)에 따라 매우 강력합니다. 물론, 서로 다른 시점과 서로 다른 유니버스에 대해 최적의 가중치는 다를 것입니다. 그러나 보시다시피 항상 잘 알려진 EF 총알 모양을 얻을 수 있습니다.

사실, 임의의 가중치가 있는 스프레드시트를 사용하는 이 직접 EF 접근 방식은 이해하기 쉽고 직설적이며 "전술적, 장기 전용" MVO의 아름다움을 보여줍니다. 그러나 특히 N 이 크고 백 테스트를 진행하면서 매월 EF(및 주어진 TV에 대한 최적의 가중치)를 계산해야 하는 경우에는 계산이 매우 매우 느릴 수 있습니다. 경험이 많은 퀀트가 수치 프로그래밍 방법을 사용하는 이유입니다.

분석적으로 목표 변동성이 주어지면 최적의 가중치 조합을 찾는 것은 제약 조건 하에서 변동성이 가중치의 2차 함수이기 때문에 2차 최적화 문제입니다(각주 2에서 설명한 대로). 전통적으로 투자자는 복잡한 2차 최적화 프로그램(제약 있음)을 사용할 수 있습니다.

³ 따라서 초과 수익률을 사용할 때 CAA와 목표 변동성을 사용하여 동일한 최적의 포트폴리오 가중치를 찾을 수 있습니다. 정상적인 수익 대신(월 무위험 이자율 제공). 그러나 MSR 솔루션은 무위험 이자율 수준이 모든 수익과 동일한 금액만큼 변경되지 않는 한 이러한 변경 사항에 대해 불변하지 않습니다. 그리고 물론 이 불변성은 주어진 달에 모든 수익이 같은 양만큼 이동했을 때만 유지됩니다. 또한 다음 항을 참조하십시오. 아래 4.

(EF의 목표 변동성에 해당하는 것과 같은) 최적의 포트폴리오 솔루션을 찾기 위해.

그러나 많은 2차 옵티마이저는 짧은 추정 이력(예: 12개월)과 많은 자산(12보다 훨씬 더 많음)의 조합에 문제가 있습니다. 그러나 더 나은 대안이 있습니다. 바로 CLA(Critical Line Algorithm)입니다.

3. Markowitz의 임계선 알고리즘

Markowitz의 임계선 알고리즘(CLA)은 제약 조건 하에서 효율적인 프론티어를 계산하는 가장 우아하고 빠른 알고리즘 중 하나입니다(Markowitz 1956 및 1959, Kwan 2007 및 Niedermayer, 2007 참조). 또한 많은 (전통적인) 2차 MVI 최적화 프로그램과 달리 자산 수 N 이 관측 수 T 보다 훨씬 클 때 효과적입니다(Nawrocki, 1996 참조).

CLA 알고리즘의 기본 아이디어는 매우 간단합니다. 수익률이 가장 높은 자산이 있는 EF의 오른쪽 상단에서 시작하고 최상의 자산을 검색하여 왼쪽의 EF를 따라갑니다(최소 분산 MV 솔루션 방향). 한 번에 하나씩 포트폴리오에서 추가(IN) 또는 빼기(OUT)합니다. 그런 다음 EF는 자산이 포트폴리오의 IN에서 OUT으로 또는 그 반대로 변경되는 소위 "코너 포인트"를 연결하는 많은 곡선("임계선"이라고 함)으로 구성됩니다. 임계선과 꼭지점 사이에서 포트폴리오에 포함된 자산은 변경되지 않고 가중치만 변경됩니다. 모든 중요한 라인(및 코너 포인트)의 컬렉션은 오른쪽 상단 포인트에서 맨 왼쪽의 MV 솔루션까지 EF를 구성합니다. 두 꼭지점 사이의 임계선은 가중치에 대해 선형이지만(그림 2와 매우 유사) EF의 (r, v) 공간에서는 곡선(포물선)입니다(그림 3 참조). 부록 A에서는 특정 월에 대해 CLA에서 생성한 EF의 몇 가지 예와 다양한 코너 포트폴리오가 있는 유니버스를 보여줍니다.

기존의 많은 MVO 옵티마이저는 $N \times N$ 공분산 행렬이 "단수"인 경우에 문제가 있습니다⁴. 공분산 행렬의 이 특이점은 자산 수 N 이 관측 수 T 보다 클 때 발생합니다. 전략적(장기적) 록백 대신 전술적(단기적) 록백을 사용하면 T 가 12로 제한됩니다(매월 관찰), 가격 모멘텀을 고려하기 위해. 이는 기존의 2차 옵티마이저에 문제를 나타낼 수 있지만

CLA.

CLA의 경우 임계 라인 세그먼트에서 최적(TV) 포트폴리오의 자산 수 $n(\leq N)$ 이 관측치 T (따라서 $n < T$)보다 작은 한 CLA에 대한 고유한 솔루션이 존재합니다. .

그리고 R에서 사용할 수 있는 CLA의 오픈 소스 버전이 없었기 때문에 우리(Ilya)는 이를 직접 코딩하고(부록 B 참조) CAA 모델용 코드를 포함하여 대중에게 공개했습니다.

CLA의 모든 꼭지점에 대한 단계적 접근 방식 때문에 작은 차원($n \ll N$)의 선형(2차 대신) 문제만 각 꼭지점에 대해 해결해야 합니다. 여기서 n 은 자산의 수입니다. 코너 포인트 주변의 포트폴리오. Excel의 CLA에 대한 설명은 Kwan(2007)을 참조하십시오. 여기에서 공매도 없음("음수가 아닌 가중치") 및 투자 제한("가중 한도")과 같은 가중치에 대한 간단한 제약 처리도 표시됩니다.

⁵ CLA를 사용하여

⁴ 역 C-1이 존재하지 않는 경우 공분산 행렬 C는 특이 행렬입니다. $T < N$ 인 경우입니다.

⁵ R의 간단한 오픈 소스 CLA 알고리즘(Ilya Kipnis가 구현)은 Kwan(2007)의 이 기사를 기반으로 합니다. 우리의 알고리즘은 우리가 아는 한 R에서 CLA(및 CAA)의 첫 번째 오픈 소스 구현입니다. 보다 최적화된(오픈 소스) CLA 구현은 Niedermayer, 2007(Fortran) 및 Bailey, 2013(Python)에서 찾을 수 있습니다.

제한된 수의 (월간) 관찰($T=12$)에도 불구하고 큰 유니버스(예: $N=39$, 아래 표시)를 처리하는 데 문제가 있습니다.

4. CAA의 속성

우리의 고전적 자산 배분(CAA) 모델은 1년 미만의 모멘텀 친화적인 록백 기간 동안 평균 분산 최적화를 적용합니다. 우리는 수익률 및 변동성 측정 척도를 연간에서 월간으로 변경할 때 MVO 모델(따라서 CAA 모델)의 최적 할당의 불변성에 대해 이미 논의했습니다. 목표 변동성에 대한 최적의 CAA 할당은 수익률의 절대 수준 과도 무관합니다. 예를 들어 다른 모든 자산 수익률에서 한 달의 T-Bill 금리를 빼는 것은 결과를 변경하지 않습니다. 모든 수익률 수준이 동일한 상수 및 주어진 목표 변동성에 대한 최적 할당에 의해 변경될 때 EF는 단순히 위 또는 아래로 이동합니다. 변경되지 않은 상태로 유지⁶. 두 가지 특성(규모 및 수준 독립성)은 CAA의 견고성에 기여합니다. (자본 배분선과 EF의 접점이 위험-수익 평면에서 EF가 높거나 낮아짐에 따라 변하기 때문에 최대 Sharpe 포트폴리오는 규모 불변이 아님을 유의해야 합니다.)

모든 자산 배분 모델의 중요한 속성은 관련 없는 대안(IIA)의 독립성입니다. 기존 유니버스 자산의 선형 조합이 포트폴리오에 추가될 때 동일한 최적의 포트폴리오 결과(r, v 측면에서)를 가져야 합니다. 따라서 SPY(의 사본)가 이미 SPY를 보유하고 있는 유니버스에 추가될 때 결과는 변경되지 않아야 합니다.⁷ Choueifaty, 2011도 참조하십시오. 상관 관계 1) 동일한 EF 및 동일한 포트폴리오 결과로 이어지므로 MVI를 기반으로 하는 모든 모델은 IIA 원칙을 준수하므로

CAA 모델을 수행합니다.

주어진 목표 변동성(TV)으로 최적의 할당에 대한 CAA/CLA 구현에 초점을 맞추지만, 종종 그렇듯이 무위험 비율이 주어지면 최대화 샤프 비율(MSR)에 해당하는 EF의 지점을 선택할 수도 있습니다. MVO로 완료했습니다. 이것을 CAA 모델의 MSR 변형이라고 합니다. 그런 다음 CAA의 이 MSR 변형의 특수 하위 모델로 가장 잘 알려진 "스마트 베타" 할당에 도달합니다. 이러한 스마트 베타 모델에는 균등 가중치(또는 $1/N$), 최소 분산(MV) 또는 최소 변동성, 전체 위험 패리티(전체 RP, ERC라고도 함) 및 나이브 위험 패리티(나이브 RP) 및 최대 다각화(MD, 참조)가 포함됩니다. 초이파티, 2011). 이들은 수익률과 공분산 행렬에 대한 특정 가정이 주어진 MVO의 일반 MSR(Maximum Sharpe Ratio) 솔루션의 특수한 경우이므로 CAA/MSR의 특수한 경우이기도 합니다. 이러한 하위 모델은 예상 수익을 자세히 지정하지 않을 때 특히 중요합니다. 아래의 CAA 구현에서와 같이 모멘텀이 사용될 때는 관련성이 적습니다. 이러한 모든 스마트 베타 모델은 무위험 비율과 무관하며 여기서는 0으로 가정합니다.

Hallerbach(2013)에서 볼 수 있듯이 EW 솔루션은 모든 (예상) 수익, 변동성 및 상관관계가 모든 자산에 대해 동일하다고 가정할 때 CAA/MSR 솔루션에 해당합니다. MV 솔루션은 모든 자산 수익률이 동일하다고 가정할 때 CAA/MSR 솔루션에 해당합니다.

수익의 수준 독립성 때문에 모든 r 에 대해 $r_i=r$ 로 CLA를 실행하면 MV가 제공됩니다.

⁶ 이 통찰력에 대해 Clarence Kwan에게 감사드립니다.

⁷ 우리의 이전 모델(Keller 2012, 2013, 2014 참조)은 SPY를 추가하면 SPY가 최적의 포트폴리오의 일부인 경우 즉, 두 배의 가중치를 부여하는 경우 단순히 SPY를 복제하기 때문에 이러한 바람직한 속성을 가지고 있지 않다는 것이 분명합니다.

과거 공분산 행렬이 주어진 솔루션. 이것은 또한 (물론) CLA에서 계산한 EF의 왼쪽 지점이기도 합니다. 우리는 모든 자산 수익률이 개별 변동성(MD)에 비례할 때 CAA/MSR의 특별한 경우로서 MD 솔루션에 도달합니다. 이는 자산에 대해 동일한 샤프 비율을 의미합니다. r_i 를 자산당 변동성 v_i 로 대체하고 CLA/MSR을 실행하면 MD 포트폴리오에 도달하게 됩니다.

또한 모든 상호 상관관계가 동일하고 모든 자산에 대한 평균 역사적 상관관계와 같다고 가정하면 전체 RP(또는 ERC, 예를 들어 Maillard 2008 참조)에 도달하는 반면 상관관계가 0이라고 가정할 때(여전히 추가 일정한 샤프 비율로) 순진한 RP에 도달합니다. 따라서 여러 스마트 베타 모델은 CAA/MSR의 특별한 경우로, 특히 수익을 위해 모멘텀을 사용하지 않을 때 찾을 수 있습니다. 그러나 이 백서의 나머지 부분에서는 MVO에 대한 추가 이점으로 CAA와 함께 모멘텀을 사용할 것입니다.

5. 데이터

39개 자산(클래스)에 대한 총 수익률 지수 기록은 여러 소스에서 수집되었습니다. 자세한 내용은 Keller(2014b)를 참조하십시오. 수익은 Global Financial Data(GFD), Kenneth French Database(FF), Barclays, MSCI, Yahoo 및 기타 과거 데이터 제공업체에서 가져왔습니다.

자산(클래스)은 크기가 증가하는 세 개의 겹치는 유니버스로 나뉩니다.

N=8 유니버스:

- | | |
|--------------------|------------------|
| • S&P 500 | • 일본 주식 • 10년 국채 |
| • EAFE | |
| • 신흥 시장 • 미국 기술 부문 | • T-Bills • 고 |
| | 수익 채권 |

N=16 유니버스:

- | | |
|-------------------|--------------------------|
| • 비내구재 • 내구재 • 제조 | • 유틸리티 • 기타 |
| • 에너지 • 기술 • 통신 | |
| 신 • 상업 • 건강 | • 10년 국채 |
| | • 30년 국채 |
| | • 미국 지방채 • 미국 회사채 • 미국 고 |
| | 수익 채권 • T-Bills |

N=39 유니버스:

- | | |
|------------------------------|-----------------------------|
| • S&P 500 | • FTSE Global ex-US • FTSE |
| • 미국 소형주 • EAFE | 선진 주식 • FTSE 신흥 시장 • 10년 국채 |
| • 신흥 시장 • 일본 주식 • 비내구 | |
| 재 • 내구재 • 제조 • 에너지 • 기술 • 통신 | • 30년 국채 |
| | • 미국 팁 |
| | • 미국 지방채 • 미국 회사채 • 미국 고 |
| | 수익 채권 • T-Bills • 국제 정부 채권 |

- 상점 • 건강

- 유틸리티

- 기타 부문 • Dow 유틸

- 리티 • Dow

- Transports • Dow

- Industrials • FTSE US

- 1000 • FTSE US 1500

- 일본 10년 만기 국채 • 원자재(GSCI) • 금

- REITs • 모

- 기지 REITs • FX (1x) • FX

- (2x) • 목재

6. 고전적 자산 배분(CAA) 모델 구현

이제 우리는 고전적인 자산 배분 (CAA) 모델의 구현을 더 자세히 설명할 것입니다.

지금쯤 분명해졌겠지만 Markowitz MVO를 기반으로 짧은 록백 기간(최대 12개월)과 공매도 제한 없음(장기 전용)이 있습니다. 목표 변동성(TV)이 주어진 경우 CLA를 사용하여 EF에 대한 최적 할당을 계산합니다.

매달 우리는 이전 1~12개월(모멘텀 효과를 내기 위해)의 (전술적) 정보를 기반으로 최적의 자산 가중치 조합을 추정하고 다음 달에 이 조합을 사용합니다. 결과적으로 시뮬레이션은 100년 분량의 데이터를 사용하므로 100년 동안 최적화를 수행했습니다. 이것은 Excel에서 Harry의 방법을 사용하여 더 큰 유니버스에서 계산적으로 다루기 어려웠을 것이지만 (최적화되지 않 * 연간 12회 재조정 = 1200은) CLA 코드를 사용하면 프로세스가 몇 초 밖에 걸리지 않았습니다.

각 최적화에 대해 우리는 다음 달 예상 수익 및 공분산에 대한 프록시로 롤링 과거 평균(수익) 및 공분산을 사용합니다. Faber(2010), Butler(2012), Antonacci(2011, 2013) 등은 사실상 모멘텀 효과가 약 1개월에서 12개월의 범위에 걸쳐 강력하다고 제안합니다. 결과적으로 우리는 최적의 공간을 확장하고 추정치를 안정화하기 위해 1, 3, 6, 12개월 동안 관찰된 모멘텀의 평균을 선택했습니다. 이는 EAA에 대한 록백 사양과도 일치합니다(Keller, 2014b).

공분산 행렬(즉, 변동성과 상관관계)의 경우 지난 12개월 동안 수익률의 과거 공분산 행렬을 사용했습니다. 우리는 또한 공분산 행렬에 대해 월별 데이터 대신 일별 데이터를 사용해 보았지만 큰 이점을 찾지 못했습니다⁸. 이것은 추가 연구를 위한 주제입니다. 모든 수익은 배당금 등을 포함한 총 수익이라는 점에 유의하십시오.

평균 분산 최적화의 구현이 최적의 포트폴리오 가중치 집합을 찾기 때문에 포트폴리오가 때때로 상당히 집중될 가능성이 있습니다. 이러한 가능성을 줄이기 위해 우리는 모든 위험 자산에 상한선(최대 가중치)을 부과하여 사후 분산을 강화했습니다. 예를 들어, 모든 자산에 대해 25% 또는 50%의 범용 캡이 있는 경우 포트폴리오에는 최소한 resp가 포함되어야 합니다. 4개 또는 2개의 자산(즉, 가중치가 0이 아닌 경우). 이러한 강제적인 다양화는 종종 사후 샤프 비율을 향상시킵니다(부록 A 참조). 우리의 기본 설정은 모든 위험 자산에 대해 25%의 상한선을 적용하고 모든 현금과 같은 자산에 대해 상한선을 적용하지 않는 것입니다(예: 100% 상한선). 결과적으로 CAA 모델은

⁸ 일별 데이터는 공분산 행렬(훨씬 더 큰 T)의 특이성을 방지할 수 있지만 가장 큰 유니버스(N=39)의 경우에도 월별 데이터 및 CLA에는 문제가 되지 않았습니다. 또한 클러스터링(자기상관)은 일일 데이터에서 더 많은 문제가 될 것입니다. 그리고 100년 동안의 데이터(금리 인상 등 포함)에 대해 우리는 월별 데이터만 마음대로 사용할 수 있습니다.

격동의 시기에 "더 안전한" 자산으로 완전히 이동합니다. 우리는 3개월 T-Bills와 USGov10y 채권을 모두 현금으로 사용할 것
이므로 모든 시뮬레이션에서 둘 다 제한이 없습니다.

모든 시뮬레이션은 1915년 1월부터 2014년 12월까지의 월별 데이터를 사용하여 수행됩니다(1년 기준, 즉 1914년 1월부터). 우리
는 항상 순진한 벤치마크로 EW(또는 1/N) 할당을 사용할 것입니다. 백 테스트별로 보고합니다.

- R(CAGR),
- D(100년 동안 월별 관측 빈도에서 최대 감소),
- V(연간 변동성),
- SR5(임계값이 5%인 샤프 비율, 따라서 $SR5=(R-5\%)/V$)
- CR5(임계값이 5%인 Calmar Ratio, 따라서 $CR5=(R-5\%)/abs(D)$),

Sharpe and Calmar Ratio의 목표 금리 5%는 100년 테스트 기간 동안 평균 300만 T-Bill 금리에 가깝습니다. 백 테스트 성능에
영향을 미치는 자유 매개 변수(자유도)의 수는 특히 개별 매개 변수 가중치가 필요한 다른 전술 모델과 비교할 때 최소입니다
(Keller 2014b 참조). 보유 또는 이진 '인 또는 아웃' 임계값의 미리 지정된 수, 그리고; 포트폴리오 자산에 대한 가중치 체계.
대조적으로, 우리의 테스트는 최적화가 자산의 수와 가중치 모두에 대해 선택하므로 네 가지 가정만 포함합니다.

- Long 만이므로 공매도 불가
- 월별 수익률 추정 기간: 1,3,6,12개월 동일, 나머지 0
- 월별 공분산 추정기간 : 최근 12개월 동일
- 테스트 유니버스의 크기 및 자산 구성: N=8, 16 및 39

대부분의 독자에게 구현을 실용적으로 만들고 솔루션의 견고성을 개선하기 위해 long-only 제약 조건이 부과되었다는 점에 유의해야
합니다(Ma(2002) 참조). 모멘텀 록백 매개변수는 알려진 유효 모멘텀 범위에 걸쳐 있고 곡선 맞춤의 가능성을 최소화하도록
선택되었습니다. 공분산 추정 범위는 공분산 행렬의 밀도를 최대화하기 위해 가장 긴 록백 기간에 해당하도록 선택되었습니다. 마지막
으로 세 가지 다양한 유니버스를 테스트하는 목표는 최적의 유니버스에서 데이터 스누핑을 피하는 것입니다. 전반적으로, 우리의 목표
는 자유 매개변수의 수를 최소화하여 곡선 적합 결과의 가능성을 광범위하게 최소화하고 샘플에서 지속적인 성능 가능성을 최대화하는
것이었습니다.

이 작은 목록에 우리는 두 개의 간단한 매개변수를 추가할 것입니다: 모든 '위험한' 자산에 대한 하나의 최대 가중치(또는 "한
도") 및 공격 또는 방어 모델에 도달하기 위한 목표 변동성(TV).

- 25%의 최대 가중치("한도")
- 목표 변동성: 10%(공격형) 및 5%(수비형 모델)

우리는 T-Bills 및 USGov10yr를 제외한 모든 자산이 "위험"하다고 가정합니다. 두 자산 모두 위험하지 않거나 "현금"으로 간주되
므로 한도가 없습니다(또는 한도가 100%임). 다른 모든 위험 자산은 입맛에 맞는 포트폴리오를 생성하고 최소한의 다각화를 시행하기
위해 25%의 기본 상한선을 갖습니다. 위험 자산의 상한선이 25%라는 것은 선택한 현금이 없을 때 포트폴리오에 최소 4개의
위험 자산이 있고 (한도가 없는) 현금이 선택되었을 때 더 많거나 적은 자산이 있음을 의미합니다.

목표 변동성은 최적화 매개변수보다 더 많은 모델 변형입니다. 또한, 부록 A에서 우리의 결과(벤치마크 $EW=1/N$ 에 대한 우월성 측면에서)가 $cap=25\%$ 의 다른 선택에 대해 매우 견고함을 보여줍니다. 특히 이전 모델(Keller 2012-2014 참조)의 모든 무료 매개변수(w_R, w_V, w_C, w_S)와 비교할 때 데이터 스누핑에 대한 자유도가 실질적으로 남아 있지 않으므로 샘플 내 (IS) 대 표본외(OS) 비교는 이전에 했던 것처럼 비교합니다(Keller, 2014b 참조). 따라서 전체 세기(1915년 1월~2014년 12월)가 단일 확장 시뮬레이션에 사용됩니다.

이제 우리는 세 가지 유니버스에 대한 결과를 제시할 것입니다: 작은($N=8$) 글로벌 다중 자산 유니버스, 미국 섹터와 채권이 포함된 중간 크기($N=16$) 유니버스, 큰($N=39$) 글로벌 멀티 자산 유니버스 유산 우주.

7. 작은(N=8) 우주

여기에서는 1914년 1월부터 2014년 12월(100년) 동안 작은(N=8) 우주에 대한 CAA의 결과를 제시합니다. 이 유니버스는 SP500, EAFE, EEM, US Tech, Japan Topix, T-Bills, US Gov10y 및 US High Yield로 구성됩니다. 모든 비현금 자산은 기본적으로 25% 상한선을 가지며 "현금" 자산 T-Bills 및 US Gov10y는 상한선이 없습니다. 벤치마크는 전체 포트폴리오의 등가중(EW 또는 1/N) 모델입니다.

공격(TV=10%) 및 방어 모델(TV=5%) 모델에 대한 수치 결과는 그림 1과 같습니다. 6과 7, 각 그림의 그래프에서. 8과 9.

TV=10%(N=8)	CAA	전자우(1/N)
아르=	12.7%	8.7%
V=	8.3%	9.2%
D=	-17.3%	-49.7%
SR5(샤프)	92.3%	40.1%
CR5(칼마르)	44.6%	7.4%

그림 6 N=8에 대한 공격(TV=10%) 결과(1915년 1월 - 2014년 12월)

TV=5%(N=8)	CAA	전자우(1/N)
아르=	10.5%	8.7%
V=	5.8%	9.2%
D=	-10.6%	-49.7%
SR5= (샤프)	94.5%	40.1%
CR5= (칼마르)	51.5%	7.4%

그림 7 N=8에 대한 방어(TV=5%) 결과(1915년 1월 - 2014년 12월)

표(그림 6 및 7)와 그래프(그림 8 및 9) 모두에서 볼 수 있듯이 CAA 성능은 특히 기본 상한인 25%를 제외하고 자유 매개변수가 부족한 점을 고려하면 우수합니다. 다소 보수적입니다(cap=100%와 같은 보다 공격적인 대안에 대해서는 부록 A 참조).

공격형(TV=10%)과 방어형(TV=5%) 모델 모두에 대해 수익률 R, 변동성 V, 최대

, Sharpes SR5 및 Calmars CR5는 모두 (EW=1/N) 벤치마크보다 상당히 우수합니다. 드로다운 D

공격 모델의 경우 수익률 R은 EW에 비해 거의 50% 향상되었으며(12.7% 대 8.7%) 변동성 V는 약간 더 적고 특히 최대 하락률 D는 거의 2/3 감소합니다(-50% 대신 -17%). EW의 경우). 가장 큰 CAA 손실은 블랙 먼데이(1987년 10월)와 제2차 세계 대전 시작(1940년 5월)에 있었습니다(그림 참조). 9. 방어적 모델의 경우에도 수익률 R은 EW(10.5% vs. 8.7%)보다 나은 반면 변동성(5.8 vs 9.2%)과 손실(-10% vs -50%)은 절반입니다. 하나

벤치마크의 다섯 번째.

수익/위험 통계의 경우 두 모델(공격 TV=10 및 방어 TV=5%)의 샤프 비율(SR5)이 EW의 두 배 이상이며, 두 최적화 모두 샤프 비율이 0.9 이상, EW의 경우 0.4 이상입니다. Calmar 비율(CR5)은 벤치마크(EW의 경우 CR5=0.07)보다 5배 이상 향상되었습니다.

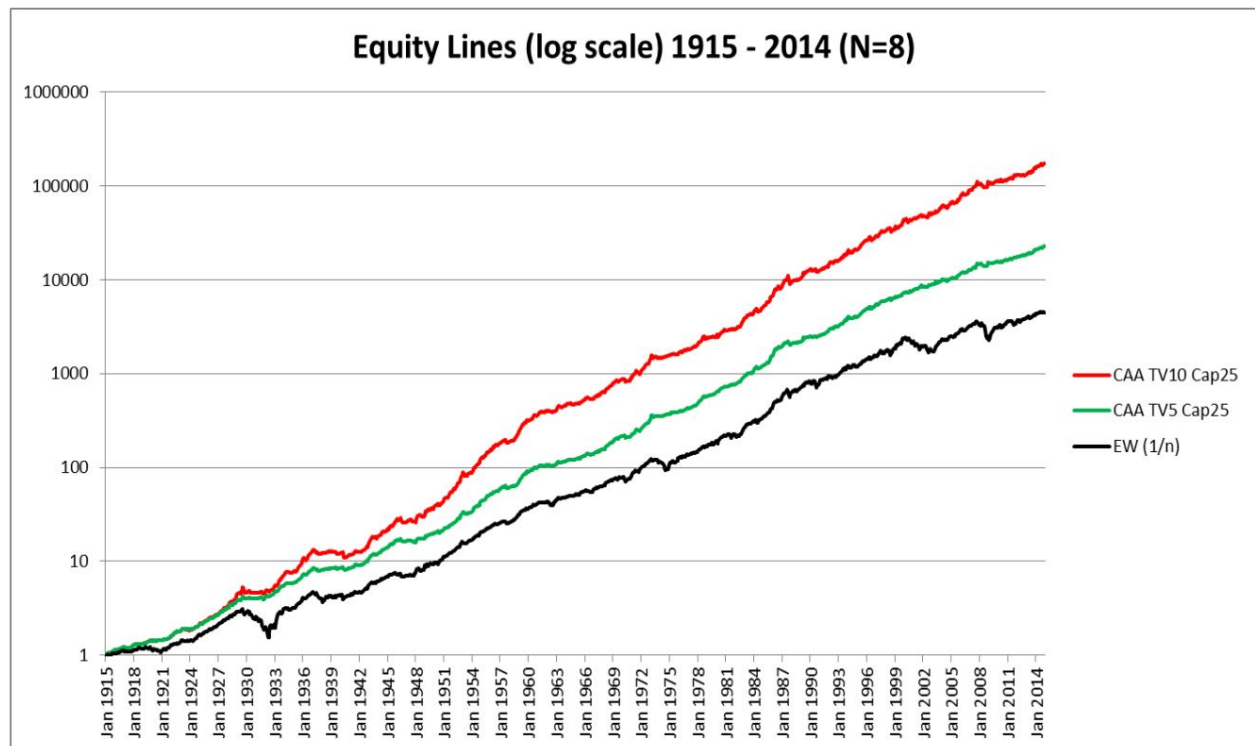


그림 8 N=8에 대한 에퀴티 라인

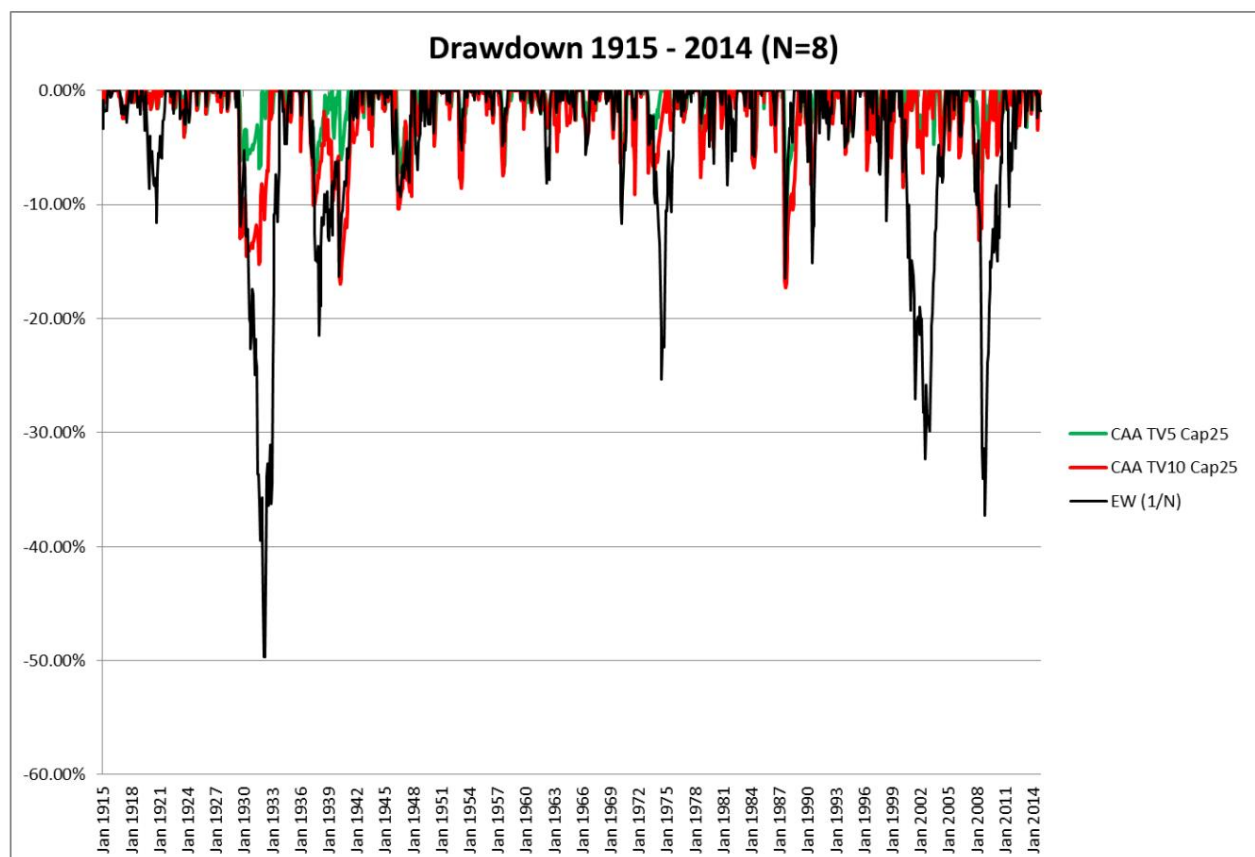


그림 9 N=8에 대한 결점

대공황이나 2008년 글로벌 금융 위기(부록 A 참조)와 같은 위기의 시기에 포트폴리오 가중치를 검토하면 CAA가 직관적으로 부분 또는 전체 현금(여기서는 T-Bills 및 USGov10y)으로 깔끔하게 전환되는 것을 볼 수 있습니다. 타임스. 이는 모멘텀 효과(단기 록백)와 현금 한도가 없는 장기 MVO의 결과일 뿐입니다.

8. 중간 크기의 유니버스(N=16)

여기에서는 1914년 1월부터 2014년 12월(100년) 동안 중간 크기(N=16) 우주에 대한 CAA의 결과를 제시합니다. 이 유니버스는 10개의 Fama/French 미국 섹터와 5개의 미국 채권(US Gov10y, US Gov30y, US Muni, US Corp, US High Yield 및 (3m) T-Bills)으로 구성됩니다. 따라서 미국 주식과 채권만 포함됩니다. 일부 자산의 경우 금세기 시작 및/또는 끝에 데이터가 누락되었습니다(Keller 2014b 참조). 모든 비현금 자산은 기본적으로 25% 한도를 가지며 두 현금 자산(T-Bills 및 US10yGov)은 한도가 없습니다. 벤치마크는 전체 포트폴리오의 증가(EW 또는 1/N) 모델입니다.

공격(TV=10%) 및 방어(TV=5%) 모델에 대한 결과는 그림에 나와 있습니다. 10 및 12, 각각. 그리고 그림에서. 11과 13.

TV=10%(N=16)	CAA	전자우(1/N)
아르=	11.20%	8.70%
V=	9.40%	11.50%
D=	19.70%	64.70%
SR5=	65.60%	32.70%
CR5=	31.30%	5.80%

그림 10 N=16에 대한 공격(TV=10%) 결과(1915년 1월 - 2014년 12월)

TV=5%(N=16)	CAA	전자우(1/N)
아르=	8.70%	8.70%
V=	5.90%	11.50%
D=	12.20%	64.70%
SR5=	62.50%	32.70%
CR5=	30.50%	5.80%

그림 11 N=16에 대한 방어적(TV=5%) 결과(1915년 1월 - 2014년 12월)

N=16 유니버스에 대한 결과에서 볼 수 있듯이 CAA에 대한 수익 R은 더 작은(N=8) 유니버스보다 적습니다. 우리는 이것이 미국만의 유니버스가 글로벌 N=8 유니버스에 비해 훨씬 덜 다양하기 때문이라고 생각합니다. 벤치마크와 관련된 N=16 CAA 모델의 개선점은 수익/위험 통계 SR5 및 CR5에서 찾을 수 있으며, 이는 공격 및 방어 모델 모두에 대해 EW보다 100% 및 500% 더 우수합니다. 최대 드로다운 D의 개선은 다시한번 인상적으로, 수비형과 공격형 모델의 경우 EW 값의 1/5, 1/3,

응답

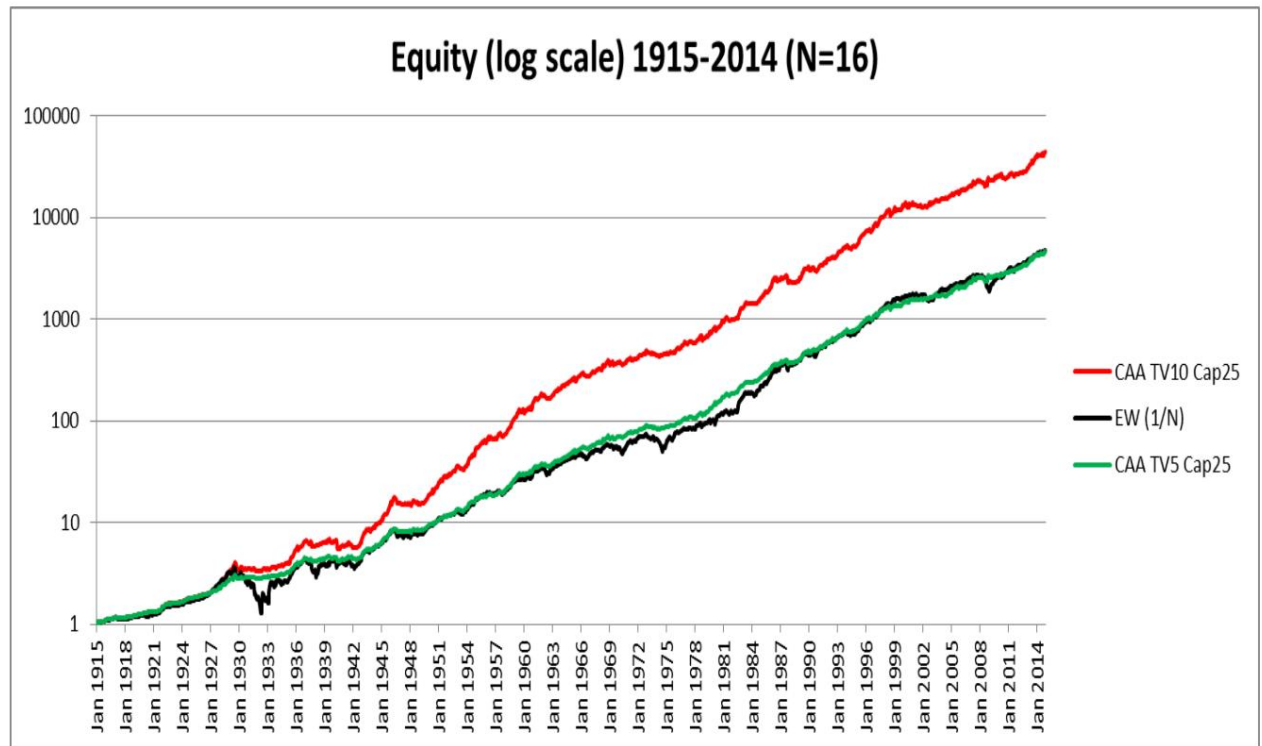


그림 12 N=16에 대한 에퀴티 라인

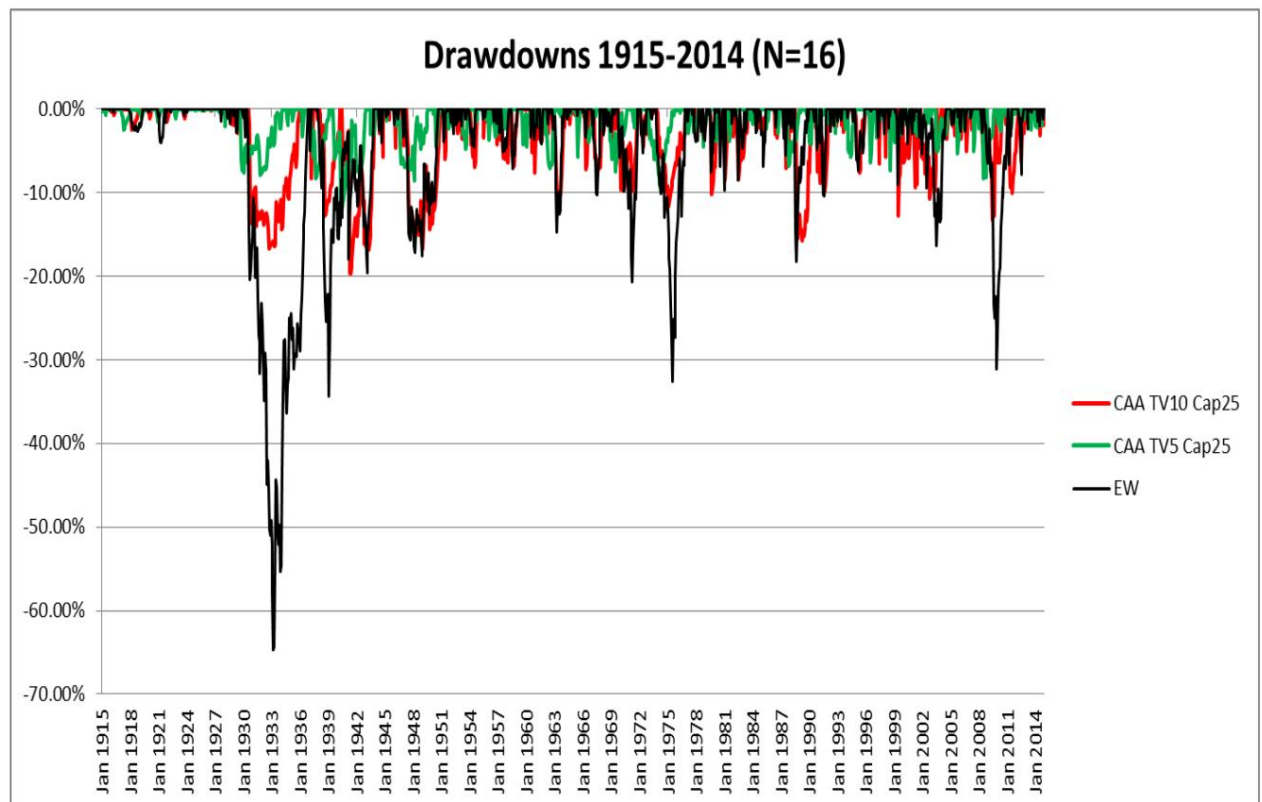


그림 13 N=16의 단점

9. 대우주(N=39)

여기에서는 1914년 1월부터 2014년 12월(100년) 동안 대규모(N=39) 우주에 대한 CAA의 결과를 제시합니다. 유니버스는 위의 두 유니버스(N=8 및 N=16)의 모든 자산과 미국 소형주 주식, GSCI, 금, 해외 채권, US TIPS, 미국 복합 리츠, 미국 모기지 리츠, FTSE US 1000/US로 구성됩니다. 1500/글로벌 전 미국/개발/EM, JapanGov10y, Dow Util/운송/산업, FX 1x/2x 및 목재. 일부 자산의 경우 금세기 시작과 끝에서 데이터가 누락되었습니다(Keller 2014b 참조). 모든 비현금 자산은 다시 기본 25% 한도를 갖게 되며 두 현금 자산(T-Bills 및 US10yGov)은 한도가 없습니다. 벤치마크는 전체 포트폴리오의 증가(EW 또는 1/N) 모델입니다.

공격 모델(TV=10%)과 방어 모델(TV=5%) 모델에 대한 결과는 Fig. 14 및 16, 각각. 그리고 무화과. 17과 18.

TV=10%(N=39)	CAA EW(1/N)	
아르=	15.40%	8.80%
V=	10.40%	10.70%
D=	22.80%	63.30%
SR5=	100.20%	35.00%
CR5=	45.80%	5.90%

그림 14 N=39에 대한 공격(TV=10%) 결과(1915년 1월 - 2014년 12월)

TV=5%(N=39)	CAA EW(1/N)	
아르=	11.80%	8.80%
V=	7.30%	10.70%
D=	15.60%	63.30%
SR5=	92.40%	35.00%
CR5=	43.30%	5.90%

그림 15 N=39에 대한 방어(TV=5%) 결과(1915년 1월 - 2014년 12월)

이러한 결과는 CLA를 사용하는 MVO가 $N \gg T$ (예: $T \leq 12$ 인 경우 $N=16$ 및 $N=39$)인 대규모 유니버스에 강력함을 분명히 보여줍니다. 결과는 인상적이며, 특히 EW 벤치마크의 거의 두 배인 공격 모델에 대한 반환 R이 인상적입니다. 샤프 비율 SR5는 1 이상(EW SR5의 거의 3배)이며 Calmar 비율(CR5=46%)은 EW 벤치마크의 7배 이상입니다. EW보다 거의 3배 작긴 하지만 D=23%는 상당합니다. V와 D가 낮고 SR5와 CR5가 거의 공격 모델만큼 우수한 방어 모델에 대해서도 유사한 관찰이 유지됩니다. 두 모델 모두 모든 면에서 EW를 분명히 이겼습니다.

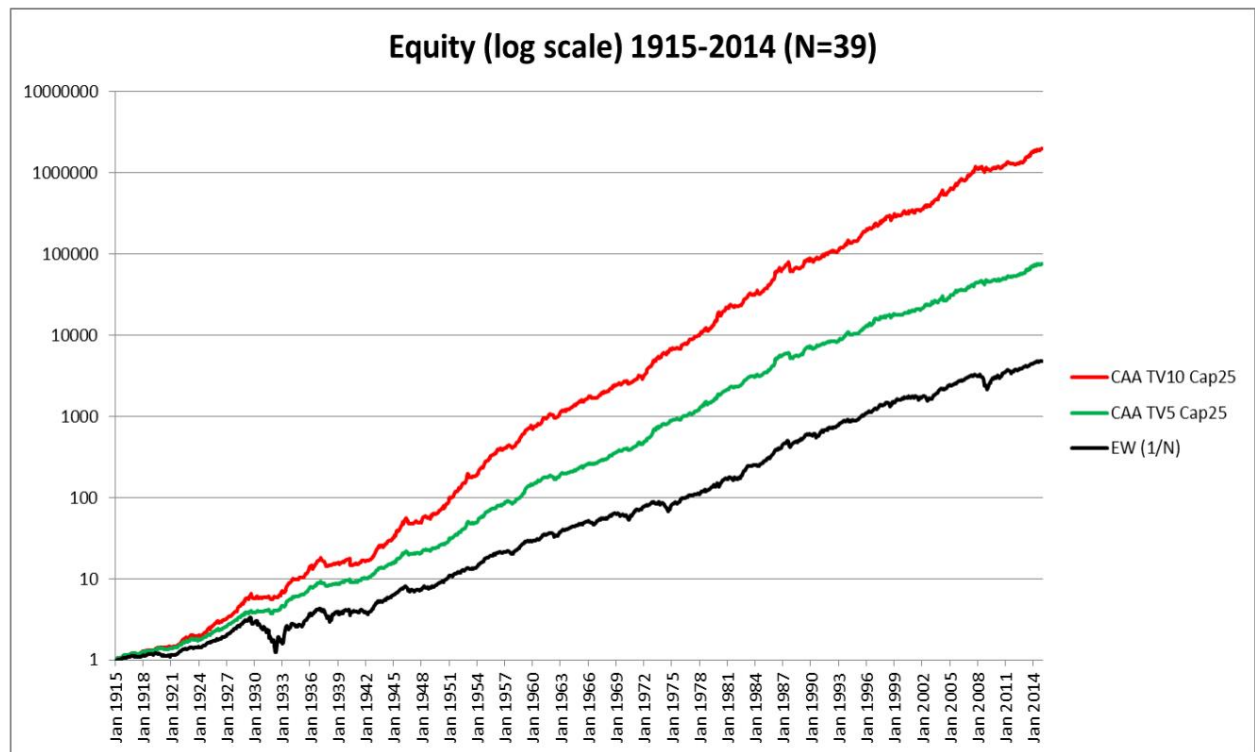


그림 16 N=39에 대한 자본선(1915년 1월 - 2014년 12월)

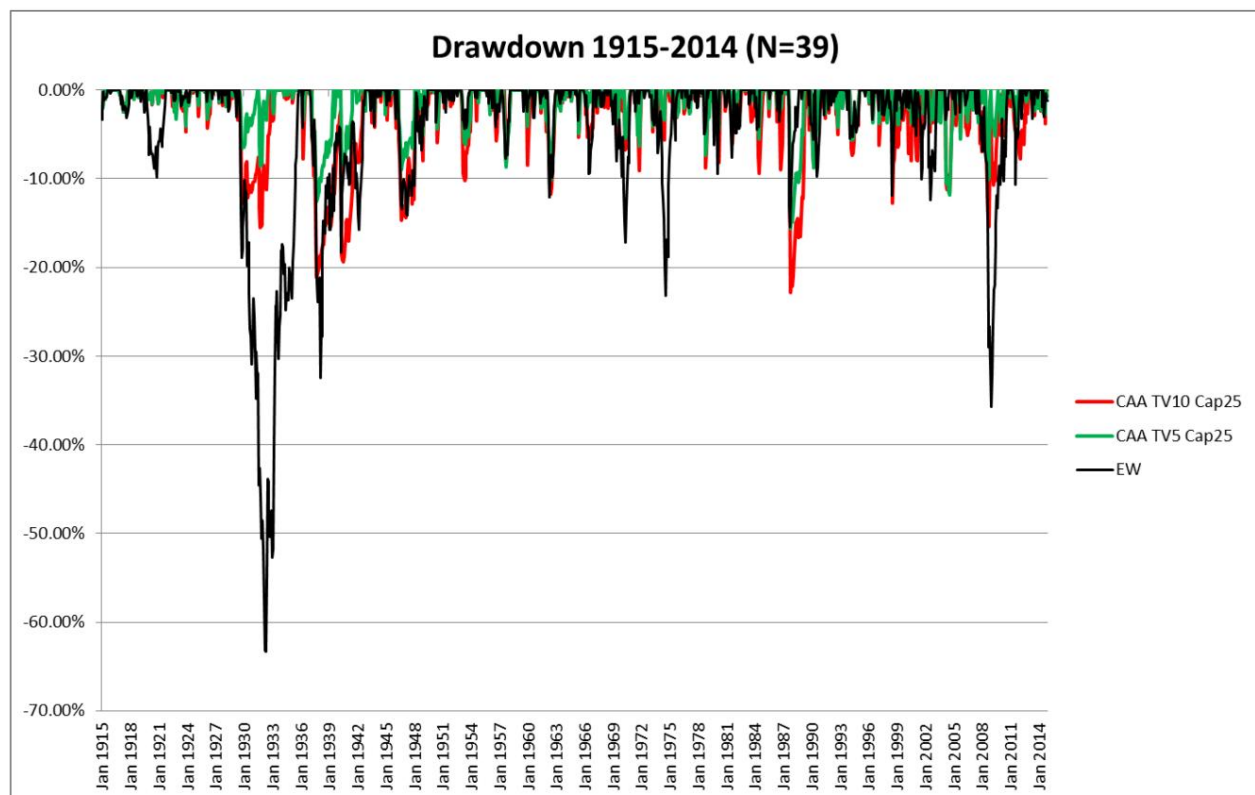


그림 17 N=39에 대한 손실(1915년 1월 - 2014년 12월)

10. 결론

첫 번째 저자가 훌륭한 미국 투자 콘테스트에 논문(Keller 2014a 참조)을 제출했을 때 심사위원장은 자신이 Top 3에 선정되지 않은 이유에 대한 질문에 대해 다음과 같은 답변을 받았습니다. 일부는 Markowitz 확인에 의해 꺼진 것 같습니다.” 따라서 Markowitz의 MPT는 투자 업계에서 더 이상 유행하지 않는 것 같습니다. 이것은 Ang(2014)의 최근 저서에서 인용한 인용문(위 참조)에 의해 확인됩니다.

현재 퀀트 "블로그 장면"에서 Markowitz는 Wesley Gray의 최근 기사(우리도 매우 존경하는 퀀트!)에서 볼 수 있듯이 그다지 인기가 없습니다.

"Markowitz는 노벨상을 수상했고 이것은 부분적으로 평균 분산 효율적인 포트폴리오를 식별하기 위한 그의 우아한 수학적 솔루션에 기반했지만 그의 아이디어가 현실 세계에 적용되었을 때 재미있는 일이 일어났습니다. 평균 분산은 제대로 수행되지 않았습니다. 노벨상 수상 아이디어가 투자자들에게 부가 가치가 없는 상황으로 해석된다는 사실은 자산 배분을 위한 최적화 방법을 고려할 때 염두에 두어야 할 사항입니다 ...복잡성은 가치와 같지 않습니다!"

이것이 투자 산업의 일반적인 주제입니다. MVO는 너무 복잡하고 불안정하며 신뢰할 수 없습니다.

동의하지 않습니다. 잘못된 것은 MVO 이론이 아니라 업계에서 MVO를 구현한 것입니다 (긴 다년간의 록백과 시끄러운 단기 판매).

특히 지난 60개월(5년) 평균 수익률을 예측 미래 수익률로 사용하는 것은 장기적인 가격 반전 이론에 직접적으로 반대됩니다.

Asness(2012, 2015)는 5년 동안의 부정적인 모멘텀을 가치에 대한 좋은 지표로 보여주었다고 강조했습니다! 그리고 우리의 구현이 공분산 행렬의 4개월 추정치와 함께 작동할 때(Harry가 보여준 것처럼) 범인은 아마도 공분산 행렬이 아니라 짧은 판매에 대한 수당과 함께 잘못된 수익 예측의 조합일 것입니다. 우리 모델을 롱온리로 제한하고 모멘텀을 사용하여 수익률을 추정하면 큰 마진으로 간주되는 모든 경우에서 쉽게 1/N을 능가합니다.

MVO를 모멘텀 및 long-only 제약 조건과 결합할 때 가장 매력적인 특징 중 하나는 데이터 스누핑에 대한 자유도가 거의 없다는 것입니다. 실용적인 구현을 위해 간단한 캡을 사용했지만 부록 A에서 볼 수 있듯이 1/N에 대한 CAA(고전적 자산 배분) 모델의 우세는 캡이 전혀 없는 것을 포함하여 선택한 캡과 독립적입니다. 오직). 우리의 록백 호라이즌은 특정 록백 값을 최적화하기보다 알려진 유효 모멘텀 공간에 걸쳐 있도록 의도적으로 선택되었습니다. 지난 12개월을 기반으로 한 간단한 과거 공분산 행렬도 마찬가지입니다.

우리의 미발표 분석(Butler, 2015)은 이러한 매개변수를 교란해도 우리의 결론에 차이가 없을 것임을 나타냅니다. 따라서 데이터 스누핑 측면에서 우리의 결과는 강력합니다. 그리고 우리가 100년이 넘는 데이터를 테스트했기 때문에 이러한 결과는 30년대(수익률 0)와 70년대 CAA 결과에서 볼 수 있듯이 금리가 상승할 때도 유효합니다(많은 사람들이 QE 이후 예상).

추가 연구를 위한 제안으로 우리는 일일 데이터로 연습을 반복하고 공매도 및 거래 비용도 허용할 계획입니다⁹. Butler(2015, 근간)는 Monte-Carlo 스타일 분석을 수행하고,

⁹ 다양한 유니버스(TV=10%/5%, 한도=25%)에 대한 CAA의 연간 매출은 대략 4(N=8의 경우)에서 7(N=39) 사이였습니다. 따라서 (오늘날의) 거래 비용이 0.1% 미만(때로는 0)인 경우 예를 들어 N=39에 대한 연간 수익률 R(CAGR)은 이러한 거래 비용을 고려한 후 최악의 경우 0.7% 낮아질 것입니다. 물론 20세기 초에 이러한 거래 비용은 이전보다 훨씬 높았습니다.

강력한 통계 테스트를 위한 결과 분포를 관찰하기 위해 무작위 유니버스 및 무작위 재조정 날짜를 포함합니다. 이 연구에서 우리는 또한 Markowitz와 모멘텀이 결합되는 한 MVO가 매우 강력한 통계적 유의성으로 1/N을 쉽게 능가하고 Equal Risk Contribution과 같은 보다 강력한 수익에 구애받지 않는 최적화를 능가한다는 결론을 내립니다.

0.1%. 예를 들어 R에서 $14.4 - 8.8 = 5.6\%$ 의 차이($N=39/TV=10/cap25\%$ 결과)는 100년 동안의 평균 거래 비용이 0.8% ($= 5.6/7$)만큼 높을 수 있음을 의미합니다.) CAA가 1/N에서 R을 잃기 전에.

부록 A. 일부 견고성 테스트

기본 캡(25%)에 대한 최상의 값을 선택하지 않았는지 확인하기 위해 그림에 표시합니다. 18 세 가지 유니버스에 대한 결과는 공격(TV=10%) 및 방어적(TV=5%) 모델. 한도=25%는 선택한 현금(한도 없음)이 없다고 가정하고 포트폴리오에 최소 4개의 자산이 있어야 함을 의미합니다. 이것은 N=8 유니버스(50%)에 대해 상대적으로 높기 때문에 실제로는 cap33 또는 cap25와 같이 약간 더 높은 한도를 선택할 수 있습니다. N=39 유니버스의 경우 기본 한도인 25%가 더 적합해 보입니다. 한도의 선택은 다각화의 최소 임계값에 대한 투자자 선호도의 함수이며 매개변수 최적화를 위해 제공되지 않습니다.

아래에서 캡 측면에서 최상의 결과를 회색으로 강조 표시합니다. 두 표에서 볼 수 있듯이 기본 상한선인 25%가 항상 수익 측면에서 최적인 것은 아닙니다. N=39, TV=10%, caps=50%에서 기본 cap=25%(R=15.4%)보다 SR5 및 CR5가 약간 더 나은 최상의 성능(R=16.3%)을 얻습니다. N=8, TV=5, 상한선=10%에 대한 최악의 성과(R=8.1%)는 EW(8.7%)보다 약간 낮지만 여전히 수익률/리스크가 훨씬 더 좋습니다(SR5는 거의 2배이고 CR5는 EW의 5배입니다.) 및 전반적으로 최고의 드로다운(D=10.7%)이 있습니다. N=16, TV5, cap10%에 대한 전체 최악의 손실률(D=23.2%)은 여전히 EW(64.7%)보다 거의 3배 더 나은 반면, Sharpe 비율(SR5=60.7%)은 여전히 EW의 거의 두 배입니다(32.7%). 최악의 Calmar 비율(N=16, TV10, 캡 10%에 대한 CR5=27.7%)은 여전히 EW보다 거의 5배 더 좋습니다. 따라서 CAA는 캡 선택(및 유니버스 및 TV 선택)과 관계없이 EW보다 훨씬 낫습니다.

상한선은 목표 변동성(TV)에 비해 실현된 사후 변동성 V에 영향을 미칩니다. 일반적으로 (작은 예외가 있지만) 상한선이 높을수록 사후 변동성 V가 높아집니다. N=8 및 TV=10%의 경우 (TV에) 가장 적합한 상한선은 상한선 50%(V=10.1%)입니다. N=16의 경우 캡 40%(V=10.1%)이고; N=39의 경우 캡 17%(V=9.8%)입니다. 이러한 결과는 더 큰 유니버스의 더 작은 캡이 전체 유니버스의 크기에 비해 여전히 고도로 집중된 포트폴리오를 허용하기 때문에 직관적입니다.

무화과에서. 19 2007-2009년 동안 N=8, TV=10%, Cap=25%에 대한 최적의 가중치를 보여줍니다. 리먼 위기(2008년 9월) 훨씬 이전에 모멘텀과 공분산의 변화로 인해 최적 할당이 어떻게 현금으로 전환되는지 주목하십시오. 또한 25%의 상한선이 어떻게 2007년에 거의 EW와 유사한(그러나 1/N은 아님) 최적의 포트폴리오로 이어져 회전율이 감소하는지 주목하십시오.

마지막으로(그림 20) EF의 네 가지 예를 보여줍니다. 코너 포인트 수의 해당 "퇴화"(2007년 7개에서 2008년 9월 2개로)가 명확해집니다. 설명은 간단합니다. 2008년 위기 전후에 마이너스 수익률을 보인 자산이 너무 많았기 때문에 플러스 수익률을 보인 제한된 수의 자산(대부분 채권, 2008년 6월 이전 일부 기술 및 EEM, 그림 19 참조)만이 수익을 올릴 수 있습니다. 포함되어 코너 포인트의 수를 제한합니다.

N8 TV10	캡10	캡17	캡25	캡33	캡40	Cap50	Cap100%	EW(1/N) 15.20%
아르=	8.80%	11.00%	12.70%	13.60%	14.10%	14.60%	8.70%	
V=	5.30%	6.90%	8.30%	9.20%	9.60%	10.10%	11.30%	9.20%
D=	-11.30%	-15.80%	-17.30%	-20.50%	-21.80%	-21.80%	-21.80%	-49.70%
SR5=	71.80%	86.20%	92.30%	93.50%	94.20%	94.90%	90.20%	40.10%
CR5=	33.50%	37.70%	44.60%	42.00%	41.60%	44.10%	46.80%	7.40%
N8 TV5	캡10	캡17	캡25	캡33	캡40	Cap50	Cap100%	EW(1/N)
아르=	8.30%	9.60%	10.50%	10.80%	10.90%	11.10%	11.00%	8.70%
V=	4.40%	5.20%	5.80%	6.10%	6.30%	6.50%	6.70%	9.20%
D=	-8.10%	-10.10%	-10.60%	-10.60%	-10.70%	-13.00%	-17.20%	-49.70%
SR5=	76.00%	88.70%	94.50%	94.80%	94.40%	95.10%	89.60%	40.10%
CR5=	40.70%	45.80%	51.50%	54.70%	55.70%	47.20%	35.10%	7.40%

N16 TV10	캡10	캡17	캡25	캡33	캡40	Cap50	Cap100%	EW(1/N)
아르=	9.80%	10.60%	11.20%	11.50%	11.70%	12.00%	12.10%	8.70%
V=	7.70%	8.70%	9.40%	9.80%	10.10%	10.40%	10.80%	11.50%
D=	-17.20%	-18.90%	-19.70%	-19.30%	-19.50%	-21.80%	-23.20%	-64.70%
SR5=	61.80%	64.20%	65.60%	66.40%	67.20%	67.10%	65.20%	32.70%
CR5=	27.70%	29.60%	31.30%	33.70%	34.60%	31.90%	30.50%	5.80%
N16 TV5	캡10	캡17	캡25	캡33	캡40	Cap50	Cap100%	EW(1/N)
아르=	8.10%	8.60%	8.70%	8.70%	8.80%	8.80%	8.80%	8.70%
V=	5.20%	5.60%	5.90%	6.10%	6.20%	6.20%	6.30%	11.50%
D=	10.70%	11.90%	12.20%	11.30%	11.40%	12.00%	12.60%	64.70%
SR5=	60.70%	63.90%	62.50%	61.60%	60.90%	61.30%	60.10%	32.70%
CR5=	29.30%	30.40%	30.50%	33.00%	32.80%	31.60%	29.90%	5.80%

N39 TV10	캡10	캡17	캡25	캡33	캡40	Cap50	Cap100%	EW(1/N) 16.30%
아르=	13.00%	14.70%	15.40%	16.00%	16.20%	15.70%	8.80%	
V=	8.80%	9.80%	10.40%	11.00%	11.40%	11.90%	12.90%	10.70%
D=	-18.70%	-21.40%	-22.80%	-23.60%	-23.50%	-23.10%	-22.70%	-63.30%
SR5=	90.40%	98.90%	100.20%	100.10%	98.20%	95.50%	83.60%	35.00%
CR5=	42.70%	45.00%	45.80%	46.70%	47.70%	49.10%	47.40%	5.90%
N39 TV5	캡10	캡17	캡25	캡33	캡40	Cap50	Cap100%	EW(1/N) 12.10%
아르=	10.50%	11.40%	11.80%	12.00%	12.10%	11.90%	8.80%	
V=	6.20%	6.90%	7.30%	7.60%	7.80%	8.00%	8.10%	10.70%
D=	-11.70%	-15.90%	-15.60%	-15.30%	-15.30%	-15.30%	-16.10%	-63.30%
SR5=	88.70%	93.30%	92.40%	92.60%	91.20%	89.00%	85.50%	35.00%
CR5=	47.40%	40.50%	43.30%	46.10%	46.40%	46.40%	43.30%	5.90%

그림 18 N=8, 16, 39 및 TV=10, 5%에 대한 다양한 캡 결과(1915년 1월 - 2014년 12월)

N8 T10	티		SP500 기술		EAFE		JapTopix EEM	
	빌	US10년 HYield						
2007-01-31 0%		0%	삼%	25%	22%	25%	0%	25%
2007-02-28 0%		0%	25%	0%	0%	25%	25%	25%
2007-03-31 0%		0%	25%	25%	0%	25%	0%	25%
2007-04-30 0%		0%	0%	25%	25%	25%	0%	25%
2007-05-31 0%		0%	0%	25%	25%	25%	0%	25%
2007-06-30 0%		0%	0%	25%	25%	25%	0%	25%
2007-07-31 0%		0%	0%	25%	25%	25%	0%	25%
2007-08-31 0%		0%	0%	25%	25%	25%	0%	25%
2007-09-30 0%		0%	0%	25%	25%	25%	0%	25%
2007-10-31 0%		0%	0%	25%	25%	25%	0%	25%
2007-11-30 0%		75%	0%	0%	0%	0%	0%	25%
2007-12-31 0%		75%	0%	0%	0%	0%	0%	25%
2008-01-31 0%	100%		0%	0%	0%	0%	0%	0%
2008-02-29 0%		75%	0%	0%	0%	0%	0%	25%
2008-03-31 0%	100%		0%	0%	0%	0%	0%	0%
2008-04-30 0%		75%	0%	0%	0%	0%	0%	25%
2008-05-31 0%		25%	25%	0%	25%	0%	0%	25%
2008-06-30 0%	100%		0%	0%	0%	0%	0%	0%
2008-07-31 0%	100%		0%	0%	0%	0%	0%	0%
2008-08-31 0%	100%		0%	0%	0%	0%	0%	0%
2008-09-30 0%	100%		0%	0%	0%	0%	0%	0%
2008-10-31 0%	100%		0%	0%	0%	0%	0%	0%
2008-11-30 8%		92%	0%	0%	0%	0%	0%	0%
2008-12-31 12%		88%	0%	0%	0%	0%	0%	0%
2009-01-31 22%		78%	0%	0%	0%	0%	0%	0%
2009-02-28 21%		79%	0%	0%	0%	0%	0%	0%
2009-03-31 22%		78%	0%	0%	0%	0%	0%	0%
2009-04-30 0%		75%	25%	0%	0%	0%	0%	0%
2009-05-31 51%		12%	25%	0%	0%	0%	0%	12%
2009-06-30 59%		0%	25%	0%	16%	0%	0%	0%
2009-07-31 46%		18%	25%	0%	0%	0%	0%	11%
2009-08-31 20%		39%	25%	0%	15%	0%	0%	0%
2009-09-30 34%		31%	25%	0%	0%	0%	0%	11%
2009-10-31 30%		32%	25%	0%	0%	0%	0%	13%
2009-11-30 0%		40%	25%	0%	20%	0%	0%	15%
2009-12-31 32%		0%	25%	0%	25%	0%	0%	18%

그림 19 N=8 및 TV=10%에 대한 가중치(2007년 1월 - 2009년 12월)

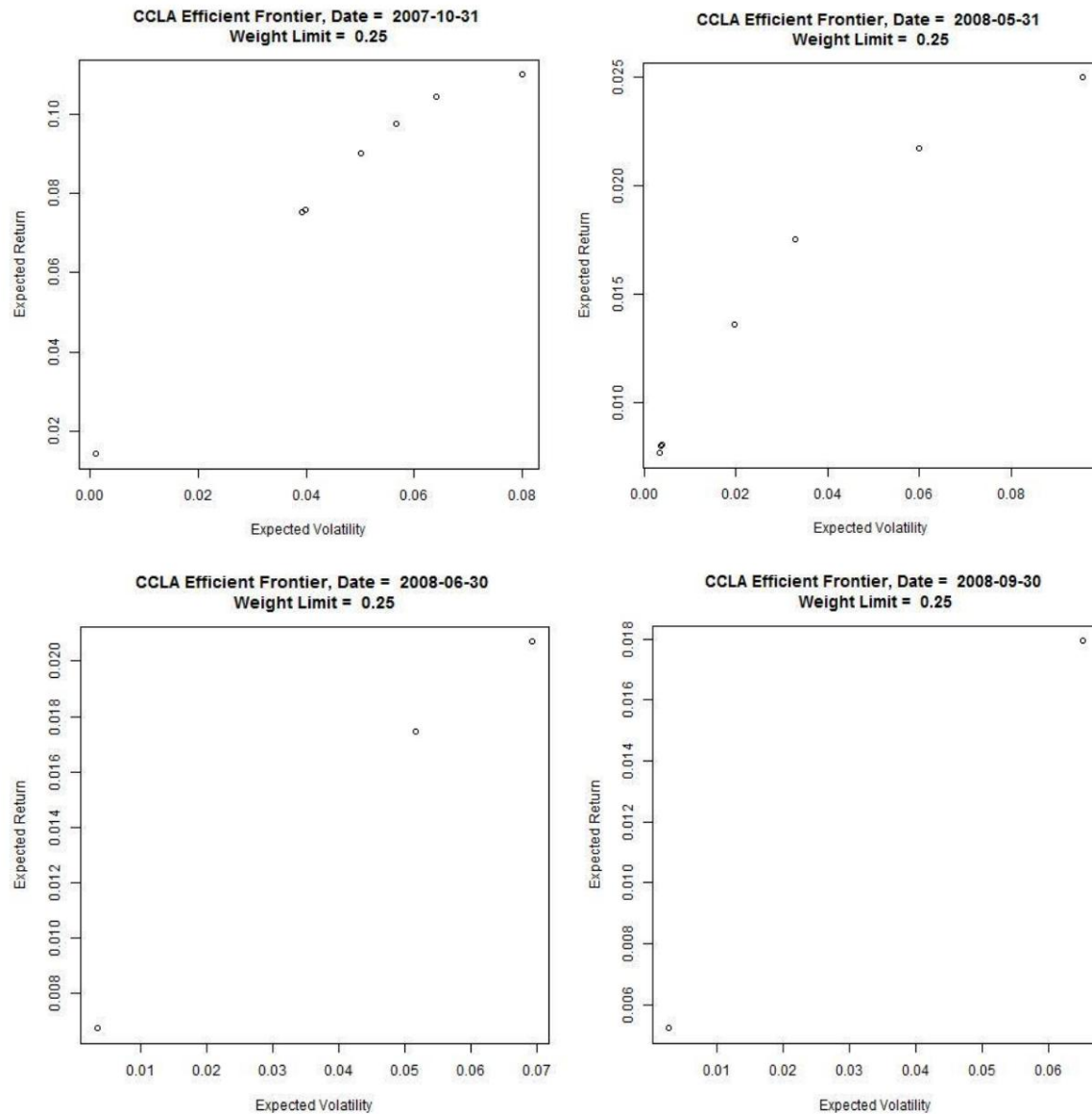


그림 20 N=8 및 TV=10%에 대한 4개의 효율적 프론티어(2007년 10월 및 2008년 5월, 6월, 9월)

부록 B. Ilya Kipnis (QuantStratTradeR10)의 CLA 코드(R)

요구(quantmod) 요구

(PerformanceAnalytics) 요구(TTR)

```
CCLA <- 함수(covMat, retForecast, maxIter = 1000,
             verbose = FALSE, scale = 252,
             weightLimit = .7, volThresh = .1) {
  if(길이(retForecast) > 길이(고유(retForecast))) { sequenceNoise
    <- seq(1:length(retForecast)) * 1e-12 retForecast <- retForecast
    + sequenceNoise }
```

```
# 원래 아웃/인/업 상태 초기화
if(length(weightLimit) == 1)
  { weightLimit <- rep(weightLimit, ncol(covMat)) }
```

```
rankForecast <- length(retForecast) - rank(retForecast) + 1 잔여 중
량 <- 1 # upStatus <- inStatus <- rep(0, ncol(covMat)) i <- 1을 할당
하기 위해 가중치의 100%를 가짐
```

```
while(remainingWeight > 0)
{ securityLimit <- weightLimit[rankForecast == i]
  if(securityLimit < residualWeight)
    { upStatus[rankForecast == i] <- 1 # 나머지 가중치를 모두 보안에 투자할 수 없는 경우 residualWeight <- 남은
      무게 - securityLimit } else { inStatus[rankForecast == i] <- 1 잔
    여 무게
      <- 0 } i <- i + 1
```

```
}
```

```
#초기 행렬(W, H, K, 항등식, 음수 항등식) covMat <-
as.matrix(covMat) retForecast <-
as.numeric(retForecast) init_W <-
cbind(2*covMat, rep(-1, ncol(covMat))) 초기화_W <-
rbind(초기화_W, c(rep(1, ncol(covMat)), 0))
H_vec <- c(rep(0, ncol(covMat)), 1)
K_vec <- c(retForecast, 0)
negIdentity <- -1*diag(ncol(init_W))
identity <- diag(ncol(init_W))
matrixDim <- nrow(init_W)
weightLimMat <- matrix(rep(weightLimit, matrixDim), ncol=ncol(covMat), byrow=TRUE)
```

```
#out 상태는 단순히 in 또는 up이 아닌 것입니다.
outStatus <- 1 - inStatus - upStatus
```

¹⁰ <https://quantstrattrader.wordpress.com/>

```

#initialize 예상 변동성/개수/전환점 데이터 구조 expVol <- Inf lambda <- 100

카운트 <- 0
TurningPoints <- list()
while(lambda > 0 & count < maxIter) {

  #숫자 알고리즘과 함께 사용하기 위한 이전 람다 및 이전 예상 변동성 oldLambda <- lambda oldVol <-
  expVol

  카운트 <- 카운트 + 1

  #compute W, A, B
  inMat <- matrix(rep(c(inStatus, 1), matrixDim), nrow = matrixDim, byrow = TRUE) upMat <-
  matrix(rep(c(upStatus, 0), matrixDim), nrow = matrixDim, byrow = TRUE) outMat <-
  matrix(rep(c(outStatus, 0), matrixDim), nrow = matrixDim, byrow = TRUE)

  승 <- inMat * init_W + 업매트 * 아이덴티티 + 아웃매트 * 부정 ID

  inv_W <- solve(W)
  modified_H <- H_vec - rowSums(weightLimMat* upMat[, -matrixDim] * init_W[, -matrixDim])
  A_vec <- inv_W %*% modified_H
  B_vec <- inv_W %*% K_vec

  # A 및 B 벡터에서 마지막 요소 제거 truncA <- A_vec[-length(A_vec)]
  truncB <- B_vec[-length(B_vec)]

  #비율 계산(Kwan.xls의 비율(1)이라고도 함) inRatio <- rep(0,
  ncol(covMat)) inRatio[truncB > 0] <-
  -truncA[truncB > 0]/truncB[truncB > 0]

  #비율을 계산합니다(Kwan.xls의 Ratio(2)라고도 함) upRatio
  <- rep(0, ncol(covMat)) upRatioIndices
  <- which(inStatus==TRUE & truncB < 0) if(length(upRatioIndices) >
  0) { upRatio[upRatioIndices] <-
  (weightLimit[upRatioIndices] - truncA[upRatioIndices]) /
  truncB[upRatioIndices]
  }

  # 람다 찾기 -- 최대 비율 maxInRatio <- max(inRatio)
  maxUpRatio <- max(upRatio)
  lambda <- max(maxInRatio,
  maxUpRatio)

  #새 가중치 계산 wts <-
  inStatus*(truncA + truncB * lambda) + upStatus * weightLimit + outStatus * 0

  #기대 수익률 및 새로운 예상 변동성 계산 expRet <- t(retForecast) %*% wts

```

```

expVol <- sqrt(wts %*% covMat %*% wts) * sqrt(스케일)

# 전환점 데이터 행을 만들고 전환점에 추가합니다. Lambda", "Exp. Vol.",
colnames(covMat))turningPoints[[count]] <-turningPoint

#변동성 임계값에 대한 이진 검색 -- 첫 번째 반복이 임계값보다 낮으면 #즉시 반환하고, 그렇지 않으면 람다가 수렴
할 때까지 이진 검색을 수행합니다. (rbind, TurningPoints) threshWts <- tail(turningPoints, 1)
return(list(turningPoints, threshWts)) }
else if(oldVol > volThresh & expVol < volThresh)
{ upLambda <- oldLambda dnLambda
<- lambda

meanLambda <- (upLambda + dnLambda)/2 동
안(upLambda - dnLambda > .00001) {

#평균 람다 계산 및 가중치 재계산, 예상 수익 및 예상 볼륨 meanLambda <- (upLambda + dnLambda)/
2 wts <- inStatus*(truncA + truncB *
meanLambda) + upStatus expRet <- t(retForecast) %*% wts expVol * weightLimit + outStatus * 0
<- sqrt(wts %*% covMat %*% wts) *
sqrt(스케일)

# 새로운 예상 볼륨이 임계값보다 작으면 평균이 하한이 됩니다. #그렇지 않으면 상한이 되어 루
프가 반복됩니다.

dnLambda <- meanLambda }
else
{ upLambda <- meanLambda } }

#이진 검색이 완료되면 해당 가중치를 반환하고 이진 검색까지 #계산된 꼭지점을 반환합니다. 모서리 점은
어디에도 사용되지 않지만 거기에 있습니다. threshWts <- cbind(count, expRet, meanLambda, expVol, t(wts))
colnames(turningPoint) <- colnames(threshWts) <- c("CP", "Exp. Ret.",
"Lambda", "Exp. Vol.", colnames(covMat))

터닝포인트[[카운트]] <- 터닝포인트 터닝포인트 <-
do.call(rbind, 터닝포인트) return(list(turningPoints,
threshWts)) }

#이진 검색이 수행되지 않는 코너 포인트에 대해서만 실행됩니다. maxInRatio] upStatus[inRatio ==
maxInRatio] <- 0 } else { upStatus[upRatio ==
maxUpRatio] <- 1 -
upStatus[upRatio == maxUpRatio] inStatus[upRatio == maxUpRatio] <- 0

```

```

} outStatus <- 1 - inStatus - upStatus }

#변동성 임계값에 도달하지 않은 경우에만 여기에 도달합니다. #충분히 낮
게 설정한 경우 실제로 발생할 수 있습니다.

threshWts <- 꼬리(turningPoints, 1)

return(목록(turningPoints, threshWts)) }

sumIsNa <- function(column)
{ return(sum(is.na(column))) }

returnForecast <- function(prices) { 예
측 <- (ROC(prices, n = 1, type="discrete") + ROC(prices, n = 3, type="discrete") + ROC(prices, n
= 6, type="discrete") + ROC(prices, n = 12, type="discrete"))/22 forecast <-
as.numeric(tail(forecast, 1)) return(forecast) }

kellerCLAFun <- function(prices, returnWeights = FALSE,
weightLimit, volThresh, uncappedAssets) {

if(sum(colnames(prices) %in% uncappedAssets) == 0)
{ stop("제한이 없는 자산이 없습니다.") }

# 가중치를 포함하도록 데이터 구조 초기화 weights <- list()

#compute returns
returns <- Return.calculate(prices)
returns[1,] <- 0 #impute 첫 번째 달은 0 ep <-
endpoints(returns, on = "months") for(i in 2:
(length(ep) - 12)) { priceSubset
<- 가격[ep[i]:ep[i+12]] #subset 가격 retSubset <-
returns[ep[i]:ep[i+12]] #subset은 assetNAs를 반환합니다
<- 적용( retSubset, 2, sumIsNa) zeroNAs <-
which(assetNAs == 0) priceSubset
<- priceSubset[, zeroNAs] retSubset <-
retSubset[, zeroNAs]

# 완벽하게 상관 관계가 있는 자산 제거 retCors
<- cor(retSubset)
diag(retCors) <- NA
corMax <- round(apply(retCors, 2, max, na.rm = TRUE), 7)
while(max(corMax) == 1) { 일
<- which(corMax == 1) 유효한
<- which(!names(corMax) %in% uncappedAssets)

```

```

toRemove <- intersect(ones, valid)
toRemove <- max(valid)
retSubset <- retSubset[, -toRemove]
priceSubset <- priceSubset[, -toRemove]
retCors <- cor(retSubset)
diag(retCors) <- NA
corMax <- 라운드(적용(retCors, 2, 최대, na.rm = TRUE), 7) }

covMat <- cov(retSubset) #공분산 행렬 계산

#닥터 켈러의 수익 예측 retForecast
<- returnForecast(priceSubset) uncappedIndex
<- which(colnames(covMat) %in% uncappedAssets) weightLims <-
rep(weightLimit, ncol(covMat))
weightLims[uncappedIndex] <- 1

cla <- CCLA(covMat = covMat, retForecast = retForecast, 스케일 = 12,
            weightLimit = weightLims, volThresh = volThresh) # CCLA 알고리즘 실행
CPs <- cla[[1]] #코너 포인트 wts
<- cla[[2]] #이진 검색 변동성 타겟팅 -- 최대 샤프 비율 골든 검색을 사용하는 경우 이 줄과 다음 줄 변경
wts <- wts[, 5: ncol(wts)] # 5월부터 끝까지
if(length(wts) == 1) { names(wts) <- colnames(covMat) }

0 <- rep(0, ncol(prices) - 길이(wts))
names(zeroes) <- colnames(prices)[!colnames(prices) %in% 이름(wts)] wts <-
c(wts, zeroes) wts <-
wts[colnames(prices)]

#가중치에 추가 wts <-
xts(t(wts), order.by=tail(index(retSubset), 1)) weights[[i]]
<- wts } weights <-

do.call(rbind, weights)

#계산 전략은 stratRets <-
Return.portfolio(returns, weights = weights) if(returnWeights)
{ return(list(weights,
stratRets)) } return(stratRets) }

kellerCLacomparison <- 함수(가격, weightLimits, volThresh,
                           uncappedAssets) {
  #initialize configurations
  configs <- list()
  for(i in 1:length(weightLimits)) {

```

```

weightLimit <- weightLimits[i]

#run CLA 함수 루프 -- 유가 증권이 수가 다르기 때문에 # 새 함수 구성 내에서 무게 제한
을 처리합니다. )

# 구성에 적합한 열 이름 생성 colnames(config) <-
paste("thresh", volThresh*100, "wLim", round(weightLimits[i], 2)*100, sep="_")

#구성에 추가 configs[[i]] <- 구
성 }

#combine configurations
configs <- do.call(cbind, configs)
return(configs) }

#plot 효율적인 프론티어
claEfPlot <- function(prices, weightLimit = .5, uncappedAssets = uncappedAssets) { returns
  <- Return.calculate(prices) returns[1,]
  <- 0 ep <-
  endpoints(prices, on = "months" )

  for(i in 2:(length(ep)-12))
  { priceSubset <- 가격[ep[i]:ep[i+12]]
    retSubset <- 반환[ep[i]:ep[i+12]]
    assetNAs <- 적용(retSubset, 2, sumIsNa)
    zeroNAs <- which(assetNAs == 0)
    priceSubset <- priceSubset[, zeroNAs]
    retSubset <- retSubset[, zeroNAs]

    retCors <- cor(retSubset)
    diag(retCors) <- NA
    corMax <- round(apply(retCors, 2, max, na.rm = TRUE), 7)
    while(max(corMax) == 1) { 일
      <- which(corMax == 1) 유효한
      <- which(!names(corMax) %in% uncappedAssets) toRemove
      <- intersect(ones, valid) toRemove <-
      max(valid) retSubset <-
      retSubset[, -toRemove] priceSubset <-
      priceSubset[, -toRemove] retCors <-
      cor(retSubset) diag(retCors)
      <- NA corMax <-
      round(apply(retCors, 2, 최대, na.rm = TRUE), 7) }

    retForecast <- returnForecast(priceSubset)
    covMat <- cov(retSubset)
  }
}

```

```

uncappedIndex <- which(colnames(covMat) %in% uncappedAssets)
weightLims <- rep(weightLimit, ncol(covMat))
weightLims[uncappedIndex] <- 1

cla <- CCLA(covMat = covMat, retForecast = retForecast, 스케일 = 12,
            weightLimit = weightLims, volThresh = 0) #CCLA 알고리즘 실행
if(!"ef" %in% dir())
{ dir.create("ef") }

jpeg(filename = paste0("ef/", as.character(tail(index(priceSubset), 1)), "_", "Frontier.jpeg")) plot(cla[[1]]
[, 2]*sqrt(12)~cla[[1]][,4], xlab="예상 변동성", ylab="예상 변동성" Return", main = paste("CCLA Efficient
Frontier, 날짜 = ", as.character(tail(index(priceSubset), 1)),
"\n무게 제한 = ", weightLimit))
dev.off() } }

```

문학

Ang, A, 2014, Asset Management: A Systematic Approach to Factor Investing (Financial Management Association Survey and Synthesis Series), Oxford University Press. Ang, A., 2012, 평균 분산 투자, SSRN 2131932 참조

Antonacci, G, 2011, Optimal Momentum: A Global Cross Asset Approach, SSRN 1833722

Antonacci, G., 2013a, Absolute Momentum: A Simple Rule-Based Strategy and Universal Trend Follow Overlay, SSRN 2244633

Antonacci, G., 2013b, 이중 모멘텀 SSRN 2042750 을 통한 위험 프리미엄 수확

Asness, CS, Moskowitz, TJ 및 Pedersen, LH, 2012, 모든 곳의 가치와 모멘텀, Working Paper nr. 80, 글로벌 시장에 관한 이니셔티브, 시카고 대학교.

Asness, CS, Frazzini, A., Israel, R. 및 Moskowitz, TJ, 2014, 팩트, 픽션 및 모멘텀 투자. Journal of Portfolio Management, 2014년 가을호(40주년 기념호); Fama-Miller 작업 문서. SSRN 2435323

Asness, CS, Frazzini, A., Israel, R. 및 Moskowitz, TJ, 2015, 팩트, 픽션 및 가치 투자 SSRN 2595747

Bailey, DH 및 López de Prado, M., 2013, 포트폴리오 최적화를 위한 중요 라인 알고리즘의 오픈 소스 구현, 알고리즘 2013, 6, 169-196(또한 SSRN 2197616)

Butler, A., Philbrick, M. 및 Gordillo, R., 2012, 적응형 자산 배분: 입문서, SSRN 2328254

Butler, A., Butler A., Keller, WJ, Philbrick M., and Gordillo M., 2015, 적응형 자산 할당을 위한 평균 분산 최적화: 통계적 검증(예정)

Choueifaty, Y., Froidure, T. 및 Reynier, J., 2011, 가장 다각화된 포트폴리오의 속성, SSRN 1895459

DeMiguel, V., Garlappi, L. 및 Uppal, R., 2009, 최적 대 순진한 다각화: 1/N 포트폴리오 전략은 얼마나 비효율적입니까? 재정 연구의 검토 22/5, pp.1915-1953

Faber, MT, 2007, A Quantitative Approach to Tactical Asset Allocation, Journal of Wealth Management, Spring 2007. 업데이트(2013) as SSRN 962461.

Faber, MT, 2010, 투자를 위한 상대적 강도 전략, 2010, SSRN: 1585517

W. 그레이, 2014, <http://blog.alphaarchitect.com/2014/12/02/our-robust-asset-allocation-raa> 솔루션/

Hallerbach, WG, 2013, 포트폴리오 위험 관리의 발전. 위험 ! 패리티?, SSRN 2259041

- Hurst, B., Ooi, YH 및 Pedersen, LH, 2012, A Century of Evidence on Trend-Following Investing, 작업 보고서, AQR Capital Management. 21
- Jacobs H., Müller, S. 및 Weber, M., 2013, 개인투자자는 어떻게 다각화해야 하는가? 대체 자산 배분 정책에 대한 실증적 평가, SSRN 1471955
- Jegadeesh, N. 및 Titman, S., 1993, 승자 매수 및 패자 매도: 주식 시장 효율성에 대한 시사점, Journal of Finance XLVIII, 65/91.
- Jurczenko, E., Michel, T. 및 Teiletche, J., 2013, 일반화된 위험 기반 투자, SSRN 2205979
- Keller, WJ 및 Van Putten, H., 2012, 일반화 모멘텀 및 유연한 자산 배분(FAA): 휴리스틱 접근 방식, SSRN 2193735
- Keller, WJ 및 Van Putten, H., 2013, 전술적 MPT 및 모멘텀: 현대 자산 배분(MAA), SSRN 2373086
- Keller, WJ, 2014a, Momentum, Markowitz 및 Smart Beta, 현대 포트폴리오 이론에 대한 전술적, 분석적, 실용적 관점, SSRN 2450017
- Keller, WJ 및 Butler, A., 2014b, A Century of Generalized Momentum; FAA(Flexible Asset Allocation)에서 EAA(Elastic Asset Allocation)로. SSRN 2543979 에서 사용 가능
- Kritzman, M., Page, S. 및 Turkington, D., 2010, 최적화 방어: 1/N의 오류, Financial Analysts Journal 66-2, pp. 31-39
- 관 CC, 2007, 포트폴리오 선택을 위한 Markowitz Critical Line 방법의 간단한 스프레드시트 기반 설명, 교육용 스프레드시트(eJSiE): Vol. 2: 이사. 3, 제2조.
- Ma, T. 및 Jagannathan, R., 2002, 대규모 포트폴리오의 위험 감소: 잘못된 제약을 부과하는 것이 도움이 되는 이유(2002년 5월). NBER 작업 문서 번호 w8922. SSRN 310469
- Maillard, S., Roncalli, T. 및 Teiletche, J., 2008, 동일 가중 위험 기여도 포트폴리오 속성에 대해, SSRN 1271972.
- 마코위츠, HM, 1952, Portfolio Selection, Journal of Finance, 7(1), 77-91.
- 마코위츠, HM, 1956, 선형 제약에 따른 2차 함수의 최적화, Naval Research Logistics Quarterly, III, 111-133.
- 마코위츠, HM, 1959, 포트폴리오 선택: 효율적인 투자 다양화. John Wiley and Sons, New York, and 1991 2nd ed., Basil Blackwell, Cambridge, MA.
- Markowitz, HM 및 P. Todd(2000), 포트폴리오 선택 및 자본 시장의 평균 분산 분석. Frank J. Fabozzi Associates, 펜실베이니아주 뉴호프
- Michaud, R. (1998), 효율적인 자산 관리: 주식 포트폴리오 최적화에 대한 실용 가이드. 옥스포드 대학 출판사.
- Moskowitz, T., Ooi, YH 및 Pedersen, LH, 2011, 시계열 모멘텀, Working Paper nr. 79, 글로벌 시장에 관한 아니셔티브, 시카고 대학교.

Nawrocki, D., 1996, 광범위한 자산에 대한 포트폴리오 분석, *Applied Economics* 28, pp. 1191-1198.

Niedermayer, A., Niedermayer, D., 2007, Markowitz의 Critical Line Algorithm 적용, 토론 논문 06-02, University of Bern, 경제학과(또한 2006 버전 SSRN 894842).

Roncalli, T., 2013, 위험 패리티 및 예산 책정 소개, Chapman & Hall/CRC Financial 수학 시리즈.

Sharpe, WF, 1963, 포트폴리오 분석을 위한 단순화된 모델, *Management Science*, Vol. 9, 2호, pp. 277-293