1차 확률적 지배를 하는
최대수익 포트폴리오 가중치의 탐색에 관한 연구
Optimizing Portfolio Weights for the First Degree Stochastic Dominance with Maximum Expected Return

류 춘 호
Choonho Ryu
홍익대학교 경영학부
ryuch@wow.hongik.ac.kr

Abstract

Unlike the mean-variance approach, the stochastic dominance approach is to form a portfolio that stochastically dominates a predetermined benchmark portfolio such as KOSPI. This study is to search a set of portfolio weights for the first degree stochastic dominance with maximum expected return by managing the constraint set and the objective function separately. An algorithm was developed and tested with promising results against Korean stock market data sets.

1. 서 론

불확실한 자산(risky assets)에 투자를 고려하는 투자자들은 가급적 수익은 높으면서 위험은 낮은 투자를 원하기에, 대부분의 투자자들은 하나의 자산에만 투자하기보다는 여러 자산에 투자를 분산시킴으로써 투자에 따른 위험을 감소시키려고 한다. 이럴 경우 가장 대표적으로 사용되고 있는 포트폴리오 구성 방법으로는 '평균-분산(mean-variance) 기준'을 들 수가 있다. 평균-분산 기준에 따르면, 투자자들이 느끼는 위험이 곧 포트폴리오 수익률의 표준편차라는 가정 하에, 포트폴리오의 기대수익이 높을수록 그리고 표준편차는 낮을수록 더 우수한 포트폴리오가 된다.

이러한 평균-분산 기준을 이용하면, 매우 드물게 일어나는 시장폭락 가능성을 투자자들이 염려하는 것과 같이 평균과 분산만으로는 실제 투자자들의 행동을 설명할 수 없을 경우, 투자자들이 느끼는 위험을 자산수익률의 표준편차만으로 설명하는 것은 부적절할 수 있다. 이러한 '평균-분산 기준'이 갖고 있는 문제를 보완할 수 있는 방법으로서 '확률적 지배(stochastic dominance)' 개념을 사용한 포트폴리오의 구성이 제시되었다. Levy(1992)는 이에 대한 연구들을 명확하게 정리하였으며, 류춘호·신성환(1997)은 확률적 지배기준이 평균-분산 기준과 비교하여 어떠한 장점이 있는지를 비교 설명하고 있다. 류춘호(1999)는 2차 확률적 지배를 고려할 경우 최적화의 대상이 되는 비선형 목적함수(nonlinear objective function)의 수학적 특성을 분석하여 전체최적성(global optimality)과 국부최적성(local optimality)을 검토하고, 이 알고리즘을 주식시장으로부터의 실제 자료에 적용하여 그 효율성을 시험하였으며, 류춘호(2003)는 1차 확률적 지배의 경우에 대해서 가중치를 최적화하는 알고리즘을 개발하였다. McNamara(1998)도 2차 확률적 지배 개념을 이용한 포트폴리오 구성을 고려하였지만, S&P500과 같은 시장지수와 무위험(risk-free)자산의 합성치를 2차 확률적으로 지배하면서 앞으로의 예상수익률이 최대가 되도록 하는 포트폴리오를 구성하기 위하여 이 문제를 선형계획법(linear programming)으로 정식화한 후 표본추출을 반복해가는 모의실험(simulation)을 통해서 포트폴리오를 구성하였다는 점에서 앞의 연구와는 다소 차이가 있다고 하겠다.

이 다음 단계는 확률적 지배를 하는 포트폴리오 중에 기대수익률이 가장 큰 것을 찾아 내는 것인데, Dentcheva & Ruszczynski(2003)는 2차 확률적 지배기준의 볼록성(convexity)과 기대수익률의 선형성(linearity)을 이용해서 이 문제를 선형계획(linear programming)모형으로 정식화하여 최적해를 구하는 방법을 제시하였다.

본 연구에서는, 류춘호(2003)가 보여 준 1차 확률적 지배기준의 확보가 매우 효율적이라는 점에 착안하여 우선은 1차 확률적 지배를 하는 가중치를

찾아내고, 일단 1차 확률적 지배를 확보한 후에는 그 속성을 유지하면서 포트폴리오의 기대수익률이 최대가 되는 곳으로 해를 움직여 가는 방법을 고려해 보고자 한다.

이 방법은 선형함수(linear function)의 형태를 가지는 기대수익률의 최대화는 물론 비선형함수(nonlinear function)의 형태를 가지는 경우(예: 포트폴리오의 분산 최소화)에도 적용할 수 있다는 점에서 선형의 경우에만 가능한 Dentcheva & Ruszczynski(2003)의 방법보다 일반적이라고 할 수 있다.

여기서는 우선 포트폴리오의 기대수익률(선형함수)을 최대화하는 것을 살펴보고 포트폴리오의 분산(비선형함수)을 최소화하는 경우에의 적용가능성도 함께 모색해보고자 한다.

2. 최적화 방법론

KOSPI를 2차 확률적으로 지배하는 포트폴리오 중 기대수익률을 최대로 하는 가중치를 찾아내는 문제는 아래와 같이 정식화할 수 있다.

$$
(\mathrm{P}_2)\quad
\begin{aligned}
\max_{\omega}\quad & E(X)=\frac{1}{T}\sum_{t=1}^{T}X_t \\
\text{s.t.}\quad & \max_{h}\int_{-\infty}^{h}\{F_x(y)-F_k(y)\}dy\le 0 \\
& \sum_{i=1}^{n}\omega_i = 1 \\
& \omega_i\in R,\quad i=1,2,\cdots,n.
\end{aligned}
$$

여기서 첫 제약식의 함수는 비선형(nonlinear)이지만 류춘호(1999)는 이를 0으로 하는 효율적인 알고리즘을 제시하였는데, 이를 이용하여 (P_2)의 최적해를 찾는 알고리즘을 개발하였다.

그런데 Dentcheva & Ruszczynski(2003)는 (P_2)의 실행가능영역이 볼록집합임을 증명하고 이 문제를 선형계획모형(LP; linear programming model)으로 정식화하였다. 그러나 이들이 제시하고 있는 LP의 규모가 매우 커서 컴퓨터의 실행시간(CPU time)이 많이 걸릴 수 있다는 단점도 함께 가지고 있다. N개의 주식에 대한 T시점의 자료를 사용한다면, 의사결정변수가 ($N + T^2$)개, 제약식이 ($T + T^2 + 1$)개나 된다. 예를 들어, 30개 주식의 일년 동안의 일별수익률 자료(260일분)의 경우, 67,630개의 변수와 67,861개의 제약식을 가진 LP문제가 된다. 주식 수의 변화에 따라 컴퓨터 실행시간이 어떻게 변화하는지, 그리고 본 논문에서 제시한 알고리즘과 어떤 차이를 보이는지에 대해서는 다음 장에서 살펴보기로 한다.

KOSPI를 1차 확률적으로 지배하는 포트폴리오 중 기대수익률을 최대로 하는 가중치를 찾아내는 문제는 아래와 같이 정식화할 수 있다.

$$
(\mathrm{P}_1)\quad
\begin{aligned}
\max_{\omega}\quad & E(X)=\frac{1}{T}\sum_{t=1}^{T}X_t \\
\text{s.t.}\quad & g(\omega)=\max_{h}\{F_x(h)-F_k(h)\}\le 0 \\
& \sum_{i=1}^{n}\omega_i = 1 \\
& \omega_i\in R,\quad i=1,2,\cdots,n.
\end{aligned}
$$

그런데 제약식의 $g(w)$는 볼록함수(convex function)도 아니고 전역에서 미분가능(everywhere differentiable)하지도 않기 때문에 알고리즘을 개발하기가 어려웠는데, 류춘호(2003)는 $g(w)$를 아래와 같이 다시 정의하여 미분가능하게 만들었다.

$$g(\omega)=\int_{-\infty}^{\infty}\{F_x(y)-F_k(y)\}^{+}dy$$

여기서 $\{F(\cdot)\}^{+}$는 함수가 0보다 크거나 같은 부분을 의미한다. 그러면 위의 문제 (P)는 아래와 같이 수정할 수 있다.

$$
(\mathrm{P})\quad
\begin{aligned}
\max_{\omega}\quad & f(\omega)=\frac{1}{T}\sum_{t=1}^{T}X_t \\
\text{s.t.}\quad & g(\omega)=\int_{-\infty}^{\infty}\{F_x(y)-F_k(y)\}^{+}dy=0 \\
& \sum_{i=1}^{n}\omega_i = 1 \\
& \omega_i\in R,\quad i=1,2,\cdots,n.
\end{aligned}
$$

여기서 $g(w)$는 볼록함수는 아니나 전역에서 미분가능하기 때문에 그 값을 0으로 만드는 w를 구하기만 하면 실행가능하며, $f(w)$는 w의 선형함수(linear function)이기 때문에 알고리즘의 개발이 한결 수월해진다. 즉, 위의 문제는 볼록집합(convex set)이 아닌 실행가능영역(feasible region) 위에서 볼록함수(convex function)인 선형함수를 최대화하는 문제가 된다.

류춘호(2003)는 $g(w)$가 볼록함수는 아니지만 그 값을 0으로 하는(즉, 1차 확률적 지배를 하는) w를 효율적으로 구할 수 있는 알고리즘을 제시하였는데, 이를 이용하여 알고리즘을 개발하고자 한다. 즉, 시작점에서 출발하여 우선 실행가능영역으로 진입을 하고나서(1차 확률적 지배를 확보하고 나서) 목적함수인 $f(w)$를 증가시키고 그 결과가 실행가능영역을 벗어나면 다시 실행가능영역을 찾아 들어가고 거기서 다시 목적함수를 증가시키는 과정을 계속 반복한다.

3. 예제

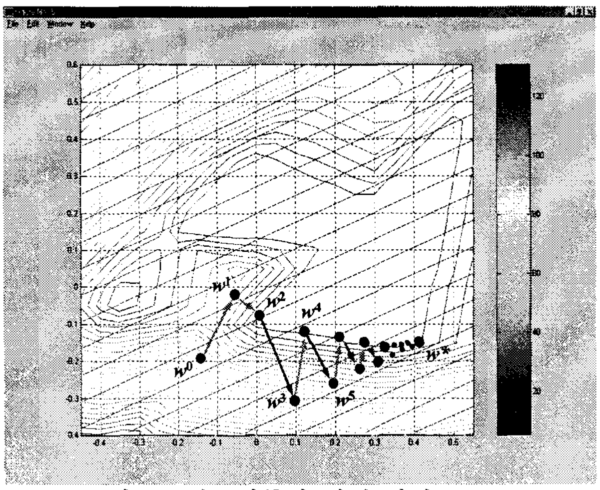
다음과 같은 조그마한 예제에 위의 알고리즘을 적용하여 어떻게 움직이는지를 살펴보고자 한다.

<표 1> 기간별 수익률

기 간	A	B	C	KOSPI
1	0.0206	0.1807	-0.0049	-0.0240
2	0.1119	-0.0816	0.0100	-0.0225
3	0.0029	0.0022	-0.0009	0.0235
4	0.0119	-0.0197	0.0396	0.0207
5	-0.0601	0.0111	0.0000	-0.0067

세 회사의 주식 A, B, C로 포트폴리오를 구성할 경우를 고려해 보자. 이때의 가중치를 각각 $\omega_1, \omega_2, \omega_3$라고 하면 $\omega_3 = 1 - \omega_1 - \omega_2$로 치환할 수가 있고, ω_1과 ω_2만 결정하면 충분하므로, $g(w)$는 (ω_1, ω_2)평면상에 나타낼 수가 있다. <그림 1>을 보면, $g(w)$를 등고선으로 표시하였는데, 볼록(convex)이 아님을 분명히 보여주고 있다. 빗금이 등간격으로 그려져 있는데 이는 목적함수인 $f(w)$의 등고선을 나타내고 있다.

본 연구에서 제시한 알고리즘은 첫 시작점 ω^0에서 출발하여 ω^1을 거쳐 ω^2에 도달하면 $g(w)$의 값이 0이 되면서 실행가능영역으로 진입하게 된다. 그 다음으로 목적함수인 $f(w)$를 증가시켜 ω^3로 이동하나 이는 실행불가능이므로 다시 실행가능한 ω^4로 이동하게 된다. 이를 지속적으로 반복해서 결국에는 ω^*에 도달하게 된다.



<그림 1> 알고리즘의 이행 과정

30개 주식의 260일의 수익률 자료에 대하여 본 연구에서 제시한 2차 확률적 지배 알고리즘을 적용하여, 반복(iteration) 횟수와 그에 따른 목적함수 및 컴퓨터 실행시간의 변화를 살펴 보면 <표2>와 같다. 여기서 f^*는 2차 확률적 지배를 하는 최대수익포트폴리오의 기댓값을 나타내며, Dentcheva & Ruszczynski(2003)의 알고리즘을 이용하여 GAMS/CPLEX로 구한 값(0.02742820)이다. CPU time은 IBM PC(Pentium III, 800MHz, 128MB RAM)를 이용한 시간이다. 으로서, 1차 확률적 지배를 하는 최대수익포트폴리오의 기댓값의 상한(upper limit)이 되는데, 이는 2차 확률적 지배가 1차 확률적 지배를 포함하는 개념이기 때문이다. $f(w)$는 반복횟수에 따른 목적함수값을 나타내며, % to f^*는 최적목적함수값에 대한 $f(w)$의 %를 나타내고 있다.

<표 2> 알고리즘의 반복횟수와 목적함수값의 변화

Iter. #	$f(w)$	% to f^*	CPU
1	-	-	0.06
3	0.00571138	20.823	0.06
20	0.02230006	81.303	0.22
50	0.02590755	94.456	0.44
100	0.02610632	95.181	0.77
250	0.02626310	95.752	1.87
500	0.02642973	96.360	3.63
1000	0.02649247	96.588	5.61
5000	0.02703619	98.571	27.96
10000	0.02729674	99.521	56.03
20000	0.02739140	99.866	112.05
30000	0.02741588	99.955	168.57
40000	0.02741832	99.964	224.81
50000	0.02742007	99.970	281.55
60000	0.02742060	99.972	337.80
70000	0.02742092	99.973	394.15
80000	0.02742104	99.974	450.45
90000	0.02742111	99.974	506.80
100000	0.02742112	99.974	565.79

이 표를 살펴보면, 이 알고리즘이 불과 100번의 반복(0.77초)만에 95%에 도달했으며 최적해 근처의 99% 이내에서 565초의 시간 대부분을 보내고 있는 걸 알 수가 있다.

4. 계산 결과

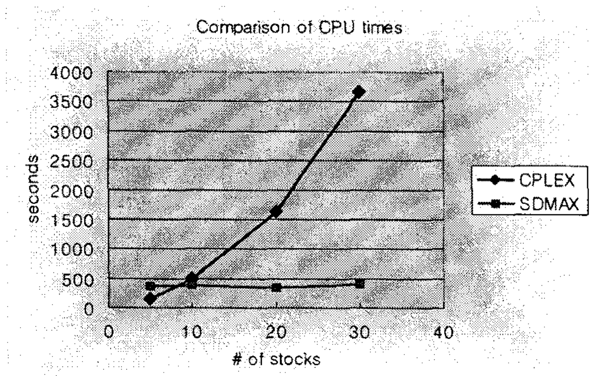
한국의 주식시장으로부터 종목별로 우량하다고 여겨지는 기업들을 골라서 총 50개 기업의 5년간의 실제 주별 주식수익률 자료에 대하여 본 연구에서 제시한 알고리즘을 적용하여 그의 효율성을 알아보았다.

<표 3>은 이들 중 임의로 5개, 10개, 20개, 30개 회사를 선택해서 260개의 주식수익률 자료에 대해서 각각의 경우에 대하여 주식 조합을 5번에 걸쳐서 임의로 골라 2차 확률적 지배 알고리즘을 적용해 본 결과의 평균치를 보여주고 있다. N은 주식의 개수, T는 주식수익률의 개수를 의미하며, 'LP problem size'는 Dentcheva & Ruszczynski(2003)의 알고리즘을 적용할 경우의 문제의 규모를 나타내는 것으로서 '제약'은 제약식의 개수를, '변수'는 의사

결정변수의 개수를, 그리고 '≠0'는 제약식에서 0이 아닌 계수의 개수를 의미한다. 'LP/CPLEX'는 이를 GAMS/CPLEX(HP/Unix workstation)로 풀었을 때의 반복횟수와 CPU time(초)을 나타내고, '2SD/FORTRAN'은 2차 확률적 지배 알고리즘을 FORTRAN(IBM PC)으로 풀었을 때의 반복횟수, 최종목적함수값의 최적목적함수값(LP)과의 비교 및 CPU time(초)을 나타내고 있다. <그림 2>에서 볼 수 있듯이, 주식의 수가 증가하면서 LP의 시간은 상당히 가파르게 증가하는 반면에, 본 연구에서 제시한 알고리즘은 큰 변화가 없는 것을 알 수 있다.

<표 3> 2차 확률적 지배 알고리즘의 계산결과

N	T	LP problem size			LP/CPLEX		2SD/FORTRAN		
		제약	변수	≠0	iter.#	CPU	iter.#	%	CPU
5	260	67862	67606	457403	32988	164.3	99659	99.96	371.6
10	260	67862	67611	796141	40938	497.6	100000	99.95	395.0
20	260	67862	67621	1442105	63639	1631.5	83369	99.94	339.3
30	260	67862	67631	2094673	77372	3674.2	96100	99.92	404.8



<그림 2> CPU time 비교

<표 4>는 동일한 자료에 대해서 2차 확률적 지배 알고리즘을 적용한 결과를 보여주고 있다. <표 3>과 다른 점 하나는 1차 확률적 지배 문제의 최적해는 LP로 구할 수가 없어서 '%'는 그 상한값인 2차 확률적 지배 문제의 최적목적함수값과의 비교를 나타내고 있다는 점이다. 즉, $f*(2SD;LP)$는 GAMS/CPLEX로 구한 2차 확률적 지배 문제의 최적목적함수값의 평균을 의미하며, $f*(1SD)$는 본 연구에서 제시한 1차 확률적 지배 알고리즘의 최종목적함수값을 보여주고 있다. 여기서도 알 수 있듯이 CPU time의 큰 변화 없이 95% 정도의 좋은 해를 얻을 수 있다는 점에서 알고리즘이 잘 작동하고 있다는 걸 알 수가 있다.

<표 4> 1차 확률적 지배 알고리즘의 계산결과

N	T	$f*(2SD;LP)$	Iter.#	$f*(1SD)$	%	CPU
5	260	0.018206	84762	0.017412	94.65	621.1
10	260	0.024855	100000	0.024345	97.90	709.2
20	260	0.021215	87521	0.020487	95.20	632.3
30	260	0.020831	94921	0.020083	95.48	686.7

5. 결 론

이상에서 우리는 변수들의 실증분포(empirical distribution)로부터 KOSPI를 1차 확률적으로 지배하는 포트폴리오 중 기대수익을 최대로 하는 가중치를 구하는 알고리즘을 개발하여 의미 있는 결과를 얻었다.

이 알고리즘은 주식투자 분야에는 물론이고, 변수들의 실증분포를 가지고 1차 확률적 지배(통상적인 확률적 지배)를 하는 가중치를 구하고자 하는 곳에도 이 알고리즘이 적용될 수 있을 것이라는 점에서 그 활용가능성도 기대해 볼 수가 있다.

향후 연구 방향으로는, 알고리즘 측면에서 다양한 자료에 대해서 이 알고리즘을 실험해 보면서 초기해의 선정, 스텝사이즈의 감소 등에 대한 개선을 모색해 보고, 이러한 최적가중치에 의해 구성되는 포트폴리오가 갖는 여러 가지 통계적 특성들을 살펴보고자 한다. 그리고 가중치가 음인 경우는 양인 경우보다 현실적으로 실행이 쉽지가 않기 때문에, 이 알고리즘이 모든 가중치가 양이라는 제약 하에서는 어떻게 수정되어야 하는지도 검토의 대상이다.

참고문헌

[1] 류춘호, "1차 확률적 지배를 하는 포트폴리오 가중치의 탐색에 관한 연구," *한국경영과학회지*, 한국경영과학회, 제28권 제1호(2003), pp.25-36.

[2] 류춘호, "2차 확률적 지배를 하는 가중치의 탐색에 관한 연구: 주식투자의 경우를 중심으로," *경영학연구*, 한국경영학회, 제28권 제1호(1999), pp.223-239.

[3] 류춘호, 신성환, "최적 포트폴리오 구성에 관한 연구," *경영연구*, 홍익대학교 경영연구소, **22**, pp.363-378, 1997.

[4] Dentcheva, D. & A. Ruszczynski, "Optimization with Stochastic Dominance Constraints," *SIAM Journal of Optimization*, **14(2)**, pp.548-566, 2003.

[5] Levy, H., "Stochastic Dominance and Expected Utility: Survey and Analysis," *Management Science*, Vol.38, No.4(1992), pp.555-591.

[6] McNamara, J. R., "Portfolio Selection Using Stochastic Dominance Criteria," *Decision Sciences*, Vol.29, No.4(1998), pp.785-801.