

XGBoost를 활용한 리스크패리티 자산배분 모형에 관한 연구

김영훈

국민대학교
비즈니스IT 전문대학원
(dudgnskr92@naver.com)

최흥식

국민대학교
비즈니스IT 전문대학원
(hschoi@kookmin.ac.kr)

김선웅

국민대학교
비즈니스IT 전문대학원
(swkim@kookmin.ac.kr)

.....

인공지능을 기반으로 한 다양한 연구들이 현대사회에 많은 변화를 불러일으키고 있다. 금융시장 역시 예외는 아니다. 로보어드바이저 개발이 활발하게 진행되고 있으며 전통적 방식의 단점을 보완하고 사람이 분석하기 어려운 부분을 대체하고 있다. 로보어드바이저는 인공지능 알고리즘으로 자동화된 투자 결정을 내려 다양한 자산배분 모형과 함께 활용되고 있다. 자산배분 모형 중 리스크패리티는 대표적인 위험 기반 자산배분 모형의 하나로 큰 자산을 운용하는 데 있어 안정성을 나타내고 현업에서 역시 널리 쓰이고 있다. 그리고 XGBoost 모형은 병렬화된 트리 부스팅 기법으로 제한된 메모리 환경에서도 수십억 가지의 예제로 확장이 가능할 뿐만 아니라 기존의 부스팅에 비해 학습속도가 매우 빨라 많은 분야에서 널리 활용되고 있다. 이에 본 연구에서 리스크패리티와 XGBoost를 장점을 결합한 모형을 제안하고자 한다. 기존에 널리 사용되는 최적화 자산배분 모형은 과거 데이터를 기반으로 투자 비중을 추정하기 때문에 과거와 실투자 기간 사이의 추정 오차가 발생하게 된다. 최적화 자산배분 모형은 추정 오차로 인해 포트폴리오 성과에서 악영향을 받게 된다. 본 연구는 XGBoost를 통해 실투자 기간의 변동성을 예측하여 최적화 자산배분 모형의 추정 오차를 줄여 모형의 안정성과 포트폴리오 성과를 개선하고자 한다.

본 연구에서 제시한 모형의 실증 검증을 위해 한국 주식시장의 10개 업종 지수 데이터를 활용하여 2003년부터 2019년까지 총 17년간 주가 자료를 활용하였으며 in-sample 1,000개, out-of-sample 20개씩 Moving-window 방식으로 예측 결과값을 누적하여 총 154회의 리밸런싱이 이루어진 백테스팅 결과를 도출하였다. 본 연구에서 제안한 자산배분 모형은 기계학습을 사용하지 않은 기존의 리스크패리티와 비교하였을 때 누적수익률 및 추정 오차에서 모두 개선된 성과를 보여주었다. 총 누적수익률은 45.748%로 리스크패리티 대비 약 5% 높은 결과를 보였고 추정오차 역시 10개 업종 중 9개에서 감소한 결과를 보였다. 실험 결과를 통해 최적화 자산배분 모형의 추정 오차를 감소시킴으로써 포트폴리오 성과를 개선하였다. 포트폴리오의 추정 오차를 줄이기 위해 모수 추정 방법에 관한 다양한 연구 사례들이 존재한다. 본 연구는 추정 오차를 줄이기 위한 새로운 추정방법으로 기계학습을 제시하여 최근 빠른 속도로 발전하는 금융시장에 맞는 진보된 인공지능형 자산배분 모형을 제시한 점에서 의의가 있다.

주제어 : 자산배분 모형, 리스크패리티, 포트폴리오 최적화, XGBoost, 추정 오차

.....

논문접수일 : 2020년 2월 12일 논문수정일 : 2020년 3월 11일 게재확정일 : 2020년 3월 15일
원고유형 : 일반논문 교신저자 : 최흥식

1. 서론

현대사회는 과학기술 및 인터넷의 발전으로 인해 점점 더 고도화되고 있다. 구글, IBM, 마이크로소프트 등 글로벌 기업들은 인공지능 개발에 참여하고 있으며 신약 개발, 의료기술, 자율주행차 등 수 많은 분야에서는 패러다임의 변화(paradigm shift)가 일어나고 있다. 금융산업 역시 인공지능의 도입을 위해 많은 연구가 진행되고 있으며 금융시장에 로보-어드바이저(Robo-advisor)를 도입한 금융상품들이 출시되고 있다. 로보-어드바이저란 Robot과 Advisor의 합성어로 재무설계나 투자 결정을 금융 공학적 알고리즘으로 자동화하여 온라인으로 제공하는 투자자문업자를 지칭한다(Sa et al., 2016). 최근 로보-어드바이저는 자산배분 모형과 결합한 형태로 발전되고 있다.

Markowitz(1952)는 이전까지 추상적으로 존재하던 분산투자의 효율성을 수학적으로 증명하여 현대 포트폴리오 이론을 체계화하였다. 평균분산 모형, 블랙리터만 모형, 리스크패리티 모형 등 다양한 자산배분 모형이 등장하였고 Brinson(1995)은 포트폴리오 성과가 종목 선택, 마켓타이밍 그리고 자산 배분 중 91.5%가 자산 배분을 통해 결정된다는 연구 결과를 발표하였다. 이처럼 오늘날 자산 배분의 중요성은 강조되고 있다.

Markowitz가 처음 제시하며 최적 자산배분 모형의 기반이 된 평균분산모형은 효율적 투자선(efficient frontier) 상에서 위험 대비 기대수익률이 높은 포트폴리오를 선택함으로써 최적의 포트폴리오를 선택하는 모형이다. 평균분산모형은 자산의 과거 수익률을 기반으로 계산된 평균, 표준편차 및 상관계수를 활용해 간단하게 계산할 수 있어 적용이 쉽다는 장점이 있다.

그러나 투자 비중에 대한 최적 해를 구하는 과정에서 타 자산 대비 기대수익률이 높고 위험구조가 다른 자산에 지나치게 많은 투자 비중이 계산되고 음의 기대수익률의 자산에는 투자 비중이 0이 되는 코너 해(corner solution)가 발생하게 된다. 이 점을 보완하기 위해 Black and Litterman(1991)은 시장 포트폴리오로부터 균형기대수익률을 도출하여 코너 해 문제를 해결하고 투자자 전망을 적용한 블랙리터만 모형을 제시하였다. 또한 Qian(2005)은 포트폴리오에 속한 자산들의 위험기여도를 동일하게 맞춰 특정 자산에 의해 위험이 과편중 되지 않도록 설계한 리스크패리티 모형(risk parity model)을 제안하였다. 오늘날 수많은 투자자들은 이와 같은 이론을 바탕으로 포트폴리오 전략을 수립하고 있다.

하지만 자산배분 모형은 최적화 과정에서 발생한 추정 오차로 인해 많은 문제점을 야기한다. Chopra and Ziemba(1993)는 모수 추정치의 작은 변화에도 최적 포트폴리오 구성에 큰 변화를 야기할 수 있다고 발표하였다. 그리고 Demiguel et al.(2009)은 자산배분 모형의 모형 오차나 추정 오차로 인해 자산배분 성과에 상당한 영향을 끼친다고 주장하였다. 국내 연구로는 Lee and Lee(2002)는 모수 추정치의 오차가 평균분산모형에 미치는 영향을 연구하여 기대수익률의 추정 오차가 공분산 추정 오차에 비해 훨씬 심각한 영향을 끼친다고 발표하였다. 추정 오차로 인한 최적 포트폴리오의 단점을 개선하기 위해 Lee(2014)는 역사적 공분산 이외의 다양한 공분산 추정방법을 제시하여 추정 오차를 줄이기 위한 모형들을 제시하였다.

본 연구에서도 공분산의 추정 오차를 줄이기 위한 새로운 방법으로 기계학습을 통한 예측으로 모형의 추정치를 대신하고자 한다. 최근 금융 분

야에도 기계학습을 적용한 다양한 사례들이 존재한다. Roh(2013)는 전통적으로 금융 시계열 데이터 분석에 많이 활용되는 ARCH(AutoRegressive Conditional Heteroscedasticity)와 GARCH(Generalized AutoRegressive Conditional Heteroscedasticity)에 ANN(Artificial Neural Network)을 결합시켜 단일 시계열보다 변동성 방향 예측에 성능을 개선한 인공지능형 금융 시계열 예측 모형을 제시하였다. Kim(2019)은 블랙리터만 모형의 투자자 전망을 애널리스트 시장 전망 대신 SVM(Support vector Machine)을 활용하여 객관화된 지능형 자산배분 모형을 제시하였다. Ru and Shin(2011)은 ANN을 활용하여 KOSPI200 지수옵션의 동적 델타 헷징 전략을 제시하였고 기존의 블랙-숄즈 기반의 동적 델타 헷징 방법론보다 비용적 측면에서 높은 개선 효과를 보였다. Ha et al.(2019)는 XGBoost(eXtreme Gradient Boosting)를 활용해 KOSPI20 주가지수 등락 예측하여 시계열 분석에 강점을 가진 LSTM(Long Short Term Memory)과 AR(Auto Regression)의 예측 결과를 비교하여 XGBoost가 금융 시계열 분석에 있어 유용성이 있음을 보여주었다. 그러나 Ha et al.의 연구 외에는 국내에서 XGBoost와 금융모형을 결합한 사례를 찾아보기 힘들다. 이에 본 연구도 기계학습 방법론 중 XGBoost과 전통적 방식의 자산배분 모형을 결합한 모형을설계하고자 한다.

최적 자산배분 모형 중 리스크패리티 모형은 대표적인 위험 기반 자산배분 모형으로 기대수익률을 추정하지 않고 오직 공분산만 추정한다. 상대적으로 추정 오차로 인한 영향이 적은 공분산만을 추정하기 때문에 리스크패리티는 평균분산모형에 비해 안정성이 높다. 또한 위험 회피적 성향을 갖는 모형으로 자산배분의 기본적인 목적과도 부합한다.

이에 본 연구는 자산배분 모형 중 리스크패리티의 추정오차를 줄여 기존에 리스크패리티가 가지는 장점을 살리고 좀 더 진보된 형태의 자산배분 모형을 제시하고자 한다.

2. 이론적 배경

2.1 자산배분 모형

포트폴리오의 위험에는 두 가지 종류가 존재한다. 하나는 체계적 위험(systematic risk)으로 포트폴리오 구성 자산들이 속한 경제 환경에 전체적으로 영향을 주는 분산 불가능한 위험을 말한다. 다른 하나는 비체계적 위험(unsystematic risk)으로 각 자산이 가지는 위험을 뜻한다. 비체계적 위험은 위험구조가 다른 자산들에 분산 투자하여 상쇄시키거나 축소, 제거가 가능하다.

자산배분 모형은 비체계적 위험을 줄이기 위한 계량 모형으로 투자자의 목적 및 성향에 따라 다양한 자산에 투자하여 투자위험을 줄이고 기대수익률을 높인다. 자산배분 모형은 Markowitz의 현대 포트폴리오 이론을 기반으로 발전하였으며 오늘날 많은 기관 투자자들이 자금 운용 목적으로 자산배분 전략을 수립하고 있다.

2.2 리스크패리티

리스크패리티 모형은 Qian(2005)이 처음 제안하였다. 자산들의 위험을 동일하게 맞추게 되어 특정 자산에 의한 위험 편중을 막을 수 있으며 투자 비중이 0이 되는 자산이 나타나지 않아 기존의 평균분산모형이 가지는 코너 해 문제가 발생하지 않는다는 장점이 있다. 또한, 자산의 위험만을 추정하므로 기대수익률 추정으로 야기될

문제들을 사전에 차단할 수 있어 실무적으로 적용하기 용이하다.

리스크패리티 모형은 포트폴리오의 구성 자산이 포트폴리오 전체 위험에 미치는 영향을 위험 기여도(risk contribution)라 정의하여 구성 자산들의 위험기여도를 동일하게 맞추는 최적의 투자 비중을 산출한다. 자산의 비중이 한 단위 변동할 때 포트폴리오의 위험이 변동하는 정도를 한계 위험 기여도(marginal risk contribution)라 하며 아래와 같이 투자 비중에 대한 포트폴리오 위험의 편미분으로 표현할 수 있다.

$$mrc_i = \frac{\partial \sigma_p}{\partial w_i}$$

mrc_i : asset(i)'s marginal risk contribution

σ_p : risk of portfolio

w_i : investment weight of asset(i)

위험기여도는 한계 위험기여도와 자산 비중을 곱하여 다음과 같이 표현할 수 있으며 자산군의 위험기여도의 합은 포트폴리오 위험과 동일하다.

$$rc_i = \frac{\partial \sigma_p}{\partial w_i} \times w_i \text{ or } mrc_i \times w_i$$

$$\sum_i^n rc_i = \sum_i^n \left(\frac{\partial \sigma_p}{\partial w_i} \times w_i \right) = \sigma_p$$

rc_i : asset(i)'s risk contribution

리스크패리티의 투자 비중 도출 과정은 공분산 행렬을 통해 산출할 수 있으며 각 자산들의 위험기여도 차이가 최소가 되는 시점의 투자 비중을 산출한다. 해당 과정은 다음과 같은 로직으로 표현할 수 있다.

Algorithm 1: Risk Parity asset allocation

Input: covariance_matrix

Output: weight

float sigma, variance, RC_diffs, asset_count

matrix wight, covariance_matrix, mrc, rc

Function RC(weight, covariance_matrix):

variance = weight.T @ covariance_matrix @ weight

sigma = $\sqrt{\text{variance}}$

mrc = 1/sigma X (covariance_matrix @ weight)

rc = weight X mrc

rc = rc / sum(rc)

return rc

Init weight = array 1/len(asset_count)

float tolerance = 1e-20

While RC_diffs < tolerance **do**

constraints = [sum(weight) = 1 and weight > 0]

mat_RC = flattening(RC(weight)-RC(weight).T)

minimize RC_diffs = $\sum \sqrt{\text{mat_RC}^2}$

〈Figure 1〉 Risk parity algorithm

2.3 XGBoost 모형

XGBoost(eXtreme Gradient Boosting)는 Chen and Guestrin(2016)이 처음 제시하였으며 의사결정나무(Decision Trees)의 부스팅(boosting) 기법을 개선한 모형이다. 부스팅이란 수많은 약한 분류기를 결합하여 강한 분류기를 만드는 앙상블(ensemble) 기법의 하나로 약한 트리 분류기를 순차적으로 실행하면서 이전 단계의 오차를 보완하는 방식으로 강한 분류기를 생성한다. 하지만 기존 부스팅 기법은 순차적 학습으로 인해 느리다는 단점을 가진다. XGBoost는 이런 부스팅의 느린 학습속도를 병렬화를 통해 개선하여 데이터 분석 경쟁 플랫폼인 캐글(<http://www.kaggle.com>)에서도 우수성을 입증하며 유명세를 떨치고 있다.

XGBoost는 트리생성 시 CART(Classification And Regression Trees)모형을 사용한다. CART 알고리즘은 의사결정나무의 가장 대표적인 모형

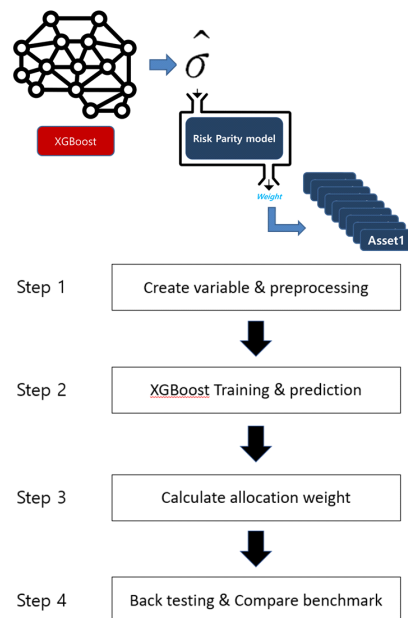
으로 범주형, 연속형 모두 학습이 가능하며 이진 분류를 통해 학습한다. 분류기준으로 불순도 척도인 지니계수(Gini index)를 사용한다. CART 방식은 트리의 모든 리프들이 모델의 최종 스코어에 연관되어 같은 분류 결과를 가지는 트리 모델 끼리도 우위를 비교할 수 있다. 생성된 트리 분류기들은 정확도 스코어를 측정해 각 순서에 따라 강한 분류기에서 약한 분류기까지 무작위로 생성하고 이를 순차적으로 개선에 나가며 강력한 트리 분류기를 생성한다.

XGBoost의 가장 큰 장점은 제한된 메모리 환경에서도 수십억 가지의 예제로 확장이 가능할 뿐만 아니라 이를 병렬처리를 통해 학습속도가 매우 빠르다는 점이다. 약한 분류기를 개선해 나가는 과정에서 탐욕적 알고리즘(greedy algorithm)을 사용하여 분류 모델들을 발견하고 분산처리를 통해 최적의 비중을 계산한다. 이 과정에서 트리 분류기의 불필요한 부분을 제거하는 가지치기를 통해 모형의 과적합화를 사전에 방지할 수 있다.

XGBoost 모형은 파라미터 조정을 통해 데이터의 특성과 모형의 용도에 맞는 모형 활용이 가능하다. 파라미터의 종류로는 크게 세분류로 나눌 수 있다. 우선 일반 파라미터로 모형의 전체적인 기능을 설정할 수 있으며 부스터의 종류, 병렬처리에 활용될 스레드 숫자를 선택할 수 있다. 두 번째는 부스팅 파라미터로 모형에서 활용되는 트리 부스터의 설정값을 제어하게 된다. 학습률(learning rate), 트리의 깊이, 트리 당 노드 개수 등 트리의 구조에 관여한다. 마지막으로 학습 과정 파라미터는 모형의 최적화 과정을 설정할 수 있다. 대표적으로 목적함수가 존재하며 데이터의 구조나 추출하고자 하는 출력 변수에 따라 다양한 설정이 가능하다.

3. 제안 모형

본 연구에서는 XGBoost로 예측한 표준편차를 활용하여 예측 공분산 행렬을 새롭게 가정하였고 이를 기존의 리스크패리티의 입력 변수인 공분산 행렬 대신 입력하여 모형을 설계하였다. 본 모형은 XGB-RP(XGBoost-Risk Parity)라 명하였으며 과거를 통한 추정이 아닌 투자 기간의 변동성을 예측함으로써 위험을 회피하고 추정오차를 줄여 모형의 안정성을 더한 모형이다. 실증 분석을 위해 국내 업종별 지수(FnGuide Universe 10: 에너지, 소재, 산업재, 경기소비재, 필수소비재, 의료, 금융, IT, 통신서비스, 유틸리티)를 활용하였다. XGBoost 학습 설계, 리스크패리티 투자 비중 산출 및 백테스팅은 파이썬(python)을 통해 구현하였다. 모형의 전체적인 구조는 <Figure 2>와 같이 설계하였다.



<Figure 2> XGB-RP's structure and process