

감성 분석이 결합된 MVO 전략: 심층 분석과 적용 가능성

2024 프로젝트

11 기 김강민

12 기 김민경 장혜영

13 기 최원준

CONTENTS

Chapter 1 : 연구 방법론	3
1.1 : 자연어 처리	3
1.2 : 감성분석	4
1.3 : MVO	5
1.4 : 몬테카를로 시뮬레이션	6
Chapter 2 : Sentiment Analysis	7
2.1 : 데이터 종류와 Sentimental 정규화	7
2.2 : Corr with KOSPI	8
2.3 : GC with KOSPI	9
Chapter 3 : 연구 및 분석	10
3.1 : 최적화 방식(MVO*Sentiment)	10
3.2 : sentimental data 추출 방식	11
3.3 : 몬테카를로를 활용한 최적 weight 판단	11
3.4 : sentimental data 변경	13
Chapter 4 : 결과	15
Chapter 5 : References	16
APPENDIX A: 2022~2024 sentimental review	17

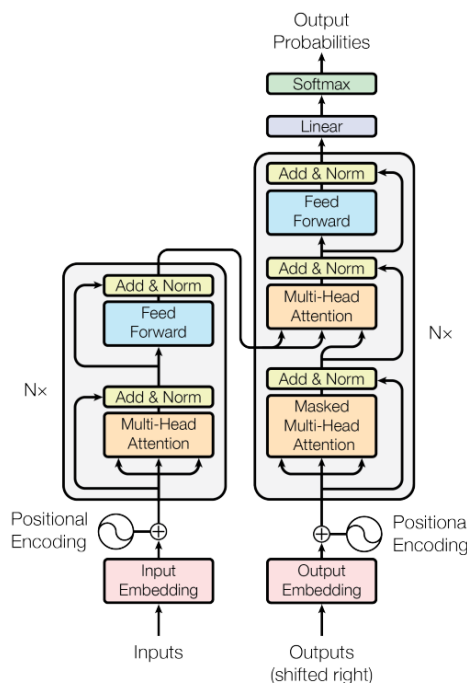
Chapter 1 : 연구방법론

1.1 자연어 처리

1.1.1 LLM(Large Language Model)

대규모 언어 모델(Large Language Model, 이하 LLM)은 자연어 처리(NLP) 분야에서 혁신적인 성능을 발휘하는 인공지능 모델이다. 이 모델은 방대한 양의 텍스트 데이터를 기반으로 훈련되어 텍스트 생성, 번역, 요약, 질의응답 등 다양한 언어 관련 작업을 수행할 수 있다. LLM의 성능과 유연성은 몇 가지 주요 특징과 구성 요소로 설명될 수 있다.

대규모 언어 모델의 핵심은 Transformer 아키텍처이다. Transformer는 2017년 Google이 발표한 모델로, 자연어 처리에서 혁신적인 도약을 가능하게 했다. 기존의 RNN이나 LSTM과는 달리, Transformer는 병렬 처리가 가능하여 훈련 속도를 크게 향상시켰다. Transformer는 인코더와 디코더의 두 부분으로 구성되며, 각 부분은 Self-Attention을 여러 번 수행하는 Multi-Head Attention 메커니즘과 피드포워드 신경망(FFN)으로 이루어져 있다.



Self-Attention 메커니즘은 문장 내의 모든 단어가 서로를 참고하여 단어 간의 관계를 학습할 수 있게 한다. 이는 문맥 이해와 의미 추론에 있어서 매우 효과적이다. 또한, Transformer는 병렬식으로 처리하기 때문에 위치 정보를

인코딩하지 못한다. 이를 보완하기 위해 Positional Encoding 을 사용하여 단어의 순서를 인식하게 한다. 이러한 구조 덕분에 LLM 은 긴 문장이나 문단 내에서도 단어 간의 복잡한 관계를 학습하고 이해할 수 있다.

1.1.2 자연어 이해(NLU)와 자연어 생성(NLG)

LLM 은 텍스트의 의미를 이해하고 새로운 텍스트를 생성하는 능력을 갖추고 있다. 자연어 이해(NLU)는 텍스트의 문맥과 의미를 파악하는 과정이다. LLM 은 방대한 양의 데이터로부터 언어의 구조와 패턴을 학습하여 텍스트의 의미를 정확히 이해할 수 있다. 이는 질문에 대한 답변을 제공하거나 텍스트의 의도를 파악하는 데 사용된다. 자연어 생성(NLG)은 주어진 정보로부터 새로운 텍스트를 만들어내는 과정이다. LLM 은 학습한 언어 모델을 바탕으로 주어진 주제나 스타일에 맞는 일관된 텍스트를 생성할 수 있다. 이는 기사 작성, 소설 창작, 마케팅 카피 작성 등 다양한 창작 활동에 활용될 수 있다. NLU 와 NLG 의 결합은 LLM 이 다양한 언어 작업을 수행하는 데 필수적이다.

1.1.3 사전 훈련과 미세 조정(Pre-training and Fine-tuning)

대규모 언어 모델은 두 단계의 훈련 과정을 거친다: 사전 훈련과 미세 조정이다. 첫 번째 단계인 사전 훈련(Pre-training)은 방대한 양의 텍스트 데이터를 사용하여 모델이 언어의 일반적인 패턴과 구조를 학습하는 과정이다. 이 과정에서 모델은 텍스트 내의 단어와 문장 간의 관계를 이해하고, 문맥을 바탕으로 의미를 파악하는 능력을 배양한다. 사전 훈련을 통해 모델은 다양한 언어 작업에서 기본적인 언어 이해 능력을 갖추게 된다. 두 번째 단계인 미세 조정(Fine-tuning)은 특정 작업이나 도메인에 맞게 모델을 추가로 훈련시키는 과정이다. 이 과정에서는 사전 훈련된 모델을 기반으로 특정 데이터셋을 사용하여 추가적인 학습을 진행한다. 이를 통해 모델은 특정 응용 분야에서 더 높은 성능을 발휘할 수 있다. 미세 조정 과정은 상대적으로 적은 데이터와 시간으로도 효과적인 모델을 생성할 수 있게 한다. 이는 다양한 산업 분야에서 LLM 을 활용할 수 있는 큰 장점으로 작용한다. 사전 훈련과 미세 조정의 결합은 LLM 이 일반적인 언어 이해와 생성 능력을 바탕으로 특정 작업에서도 높은 성능을 발휘할 수 있게 하는 중요한 요소이다.

1.2 감성 분석(Sentiment Analysis)

감성 분석(sentiment analysis)은 일반적으로(NLP: Natural Language Processing), 텍스트 마이닝 (text mining), 기계 학습(ML: Machine Learning) 등의 기술을 사용하여 문장 또는 문서에서의 표현이 긍정, 부정 혹은 중립인지를 판별하는 기술이다. 감성 분석은 일반적으로 텍스트 데이터 분석에 활용한다. 감성 분석의 기본적인 접근 방법은 '좋아요', '최고야' 등의 긍정적 어휘와 '싫어요', '별로', '슬픈' 등의 부정적 어휘를 극성으로 분류하여 텍스트의 패턴을 정량화한 후 그에 따라 데이터를 분류하고 분석하는 것이다. 어휘의 극성 분류는 사람이 평가한다. 예를

들어 긍정 혹은 부정의 정도를 '-10 ~ +10'으로 정량화하여 데이터 세트를 구축할 수 있다. 이후 측정된 데이터 세트를 이용하여 통계적 분석 방식 혹은 SVM 과 같은 머신러닝 기법으로 입력된 텍스트 데이터의 극성을 분류한다. LLM 을 이용하면 데이터세트를 구축하지 않고도 사전학습된 데이터를 기반으로 문장을 긍정, 중립, 부정으로 분류할 수 있다.

1.3 MVO

1.3.1 MVO 정의와 개념

MVO 는 포트폴리오 이론의 핵심 개념 중 하나로, 투자자의 위험 회피 성향을 고려하여 주어진 기대수익률을 최대화하거나 주어진 리스크 수준에서 수익률을 최대화하는 포트폴리오를 구성하는 방법이다. MVO 는 포트폴리오의 기대 수익률과 리스크(변동성)를 동시에 고려하며, 이 두 가지 요소 간의 균형을 맞추기 위해 다음의 수학적 최적화 문제를 푼다.

1.3.2 MVO 수학적 모델

MVO 의 핵심은 주어진 자산들의 수익률과 리스크(공분산 행렬)를 활용해 포트폴리오의 리스크를 최소화하거나 수익률을 최대화하는 것이다. 이를 위해 사용하는 공식은 다음과 같다.

$$\text{Minimize: } \frac{1}{2}w^T \Sigma w - \lambda \mu^T w$$

w : 각 자산에 대한 투자 비중의 벡터

Σ : 자산 간의 공분산 행렬

μ : 각 자산의 기대 수익률 벡터

λ : 투자자의 위험 회피 성향을 나타내는 조정 변수

제약조건으로는 모든 자산의 투자 비중의 합이 1 이어야 하고, 각 자산의 투자 비중은 음수가 될 수 없다.

1.3.3 MVO 의 한계

1) 입력 데이터의 민감성

MVO 의 경우, 자산의 기대 수익률과 공분산 행렬에 크게 의존한다. 이러한 데이터는 과거의 수익률을 기반으로 추정되며, 미래의 성과를 정확하게 예측하지 못할 수 있다. 따라서 예측의 오류가 있을 수 있다. 과거 데이터가 미래의 변동성을 정확히 반영하지 못할 경우, 최적화된 포트폴리오가 실제 시장 조건에서 부적합할 수 있다. 또한

데이터 왜곡의 문제가 있을 수 있다. 특정 사건으로 인해 데이터가 왜곡될 수 있어 최적화 결과에 부정적인 영향을 미칠 수 있다.

2) 코너 솔루션 문제

MVO는 종종 특정 자산에 과도하게 높은 비중을 할당하는 '코너 솔루션' 문제를 일으킬 수 있다. 이는 분산 투자 원칙에 반하며, 포트폴리오의 리스크가 특정 자산에 집중되어, 해당 자산의 변동성이 포트폴리오의 성과에 영향을 미칠 수 있다. 또한 자산간의 상관관계를 적절히 활용하지 못해, 분산 투자의 이점을 충분히 살리지 못하게 된다.

3) 리스크와 수익률 간의 단순한 가정

MVO는 자산의 기대 수익률과 리스크 간의 관계를 단순화하지만, 실제 금융시장에서는 더 복잡한 상호작용이 존재하기에 괴리가 존재한다.

1.4 몬테카를로 시뮬레이션(Monte Carlo Simulation)

몬테카를로 시뮬레이션은 무작위 샘플링을 통해 복잡한 시스템이나 과정을 모형화하고 분석하는 기법이다.

몬테카를로 시뮬레이션의 기본 아이디어는 다음과 같다.

- 1) 입력 변수의 확률 분포를 정의한다.
- 2) 해당 확률 분포에 따라 무작위로 샘플을 생성한다.
- 3) 생성된 샘플을 바탕으로 모델을 여러 번 반복 실행하여 결과를 도출한다.
- 4) 결과를 분석하여 통계적 특성을 추정한다.

본 프로젝트에서는 몬테카를로 시뮬레이션을 모델에 가장 최적화된 weight(가중치) 값을 설정하기 위한 방법으로 채택했다. 초기 포트폴리오의 비중을 설정하고, 수익률 분포를 기반으로 범위를 설정했다. 무작위로 weight 값을 생성해 가장 좋은 포트폴리오의 수익률을 도출해내는 weight 값을 선택할 수 있도록 적용해 코드를 작성했다.

Chapter 2 : Sentiment analysis

2.1 데이터 설명

우리가 실험에 사용할 데이터는 총 3 가지 종류의 데이터이다.

첫번째 sentimental data 는 블룸버그 서비스에서 제공하는 한국어 뉴스 사이트에서 받아온 기사 데이터이다.

(<https://www.bloomberg.co.kr/blog/category/news/>) 주로 해외 매크로 지표(PCE, 금리인하 등)뉴스를 다루고 있으며, 하루에 총 5 개의 주요 뉴스를 정리해서 올려주고 있다. 기사는 사이트에서 제공하는 2020.11.17~2024.06.05 일간의 데이터를 수집하였으며, 총 4045(809*5)개의 기사를 기반으로 추출하였다.

두번째 sentimental data 는 한국 경제 사이트에서 제공하는 코스피 태그 데이터이다.

(<https://www.hankyung.com/tag/%EC%BD%94%EC%8A%A4%ED%94%BC>) 주로 코스피와 직접적인 관련이 있는 데이터 ('코스피, 외국인 매도에 약세', '코스피 2800 아래서 출발' '자본시장법 개정안 발의' 등)를 다루고 있으며, 하루에 올라오는 기사의 개수는 제각각 다르다. 위 데이터는 첫번째 데이터와 다르게 코스피에 집중적으로 관련이 있는 데이터만 모아놓았다는 특징이 있다. 기사는 사이트에서 제공하고 있는 2016.07.08 ~ 2024.05.31 까지 총 7583 개의 기사를 기반으로 추출하였다.

세번째 sentimental data 는 매일경제 사이트에서 제공하는 코스피를 검색하였을 때 나오는 데이터이다. 이는 본문이나 기사 제목에 코스피가 들어가는 모든 종류의 데이터('우크라 살상무기 지원 검토', '밸류업 업은 중간배당', '원금 보장 ELB' 등)를 다루고 있으며, 한국 경제에서 제공하는 코스피 태그 데이터 보다 많은 데이터를 다루고 있지만 다소 노이즈와 같은 기사도 포함되어 있다. 기사는 2016-03-22 ~ 2024.06.05 까지 총 25577 개의 기사를 기반으로 추출하였다.

이렇게 포함된 sentimental data 는 정규화 과정을 거쳐서 사용하였다. 정규화 과정의 기본 개념은 다음과 같다.

$$\text{monthly sentimental score} = \frac{\sum \left(\frac{\sum (\text{daily senti score})}{\text{num daily news}} \right)}{\text{num monthly news}}$$

$$\text{min_senti_score} = \min (\text{monthly sentimental score}_{12} \sim \text{monthly sentimental score}_1)$$

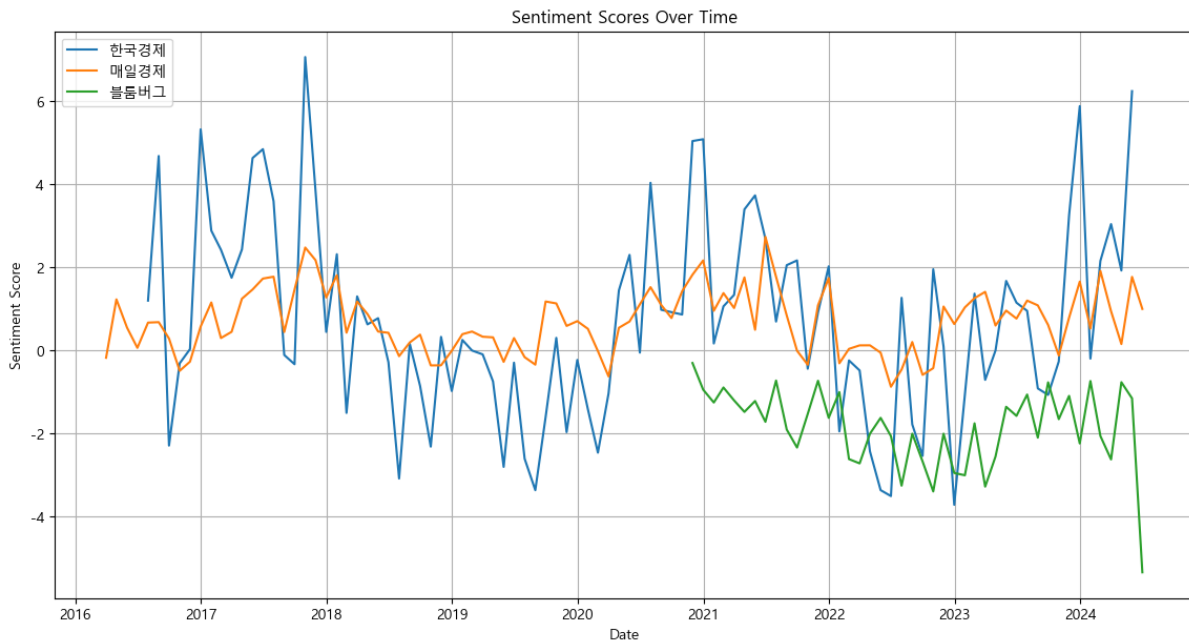
$$\text{max_senti_score} = \max (\text{monthly sentimental score}_{12} \sim \text{monthly sentimental score}_1)$$

$$\text{adjusted senti score} = 0 : \text{if } (\text{senti score} < \text{min_senti_score})$$

$$\text{adjusted senti score} = 1 : \text{if } (\text{senti score} > \text{max_senti_score})$$

$$\text{adjusted senti score} = (\text{senti score} - \text{min senti score}) / (\text{max senti score} - \text{min senti score})$$

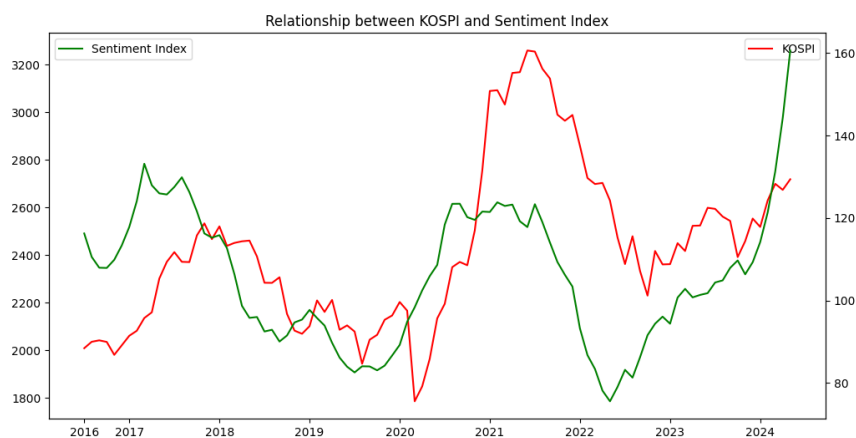
이렇게 min max score 를 통해 정규화된 sentimental 의 그래프는 다음과 같다.



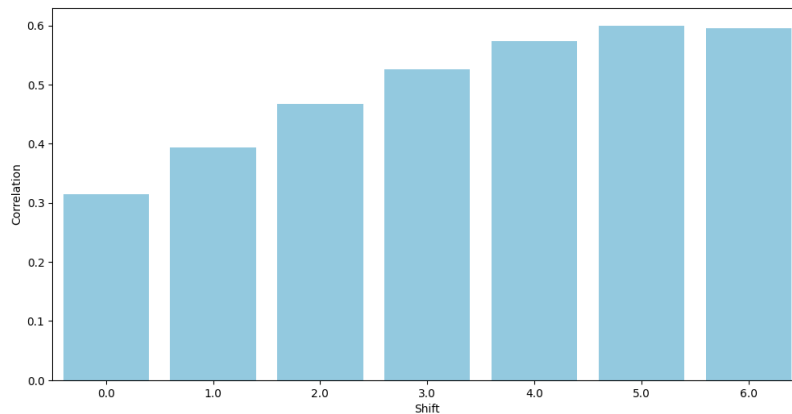
2.2 Corr with KOSPI

본 연구에서는 2016 년 7 월부터 2024 년 5 월까지 매일경제에 코스피를 검색하여 나오는 기사의 제목과 본문, 그리고 Bloomberg 한국어 뉴스 Archives 를 수집하였다. 감성 분석은 한국어 금융 도메인 특화 LLM 인 KR-FinBERT 를 활용하여 감성분석을 진행하였다. 긍정으로 분류된 문장은 1, 중립은 0, 부정은 -1 로 만든 후, 월별로 감성 지수를 산출하였다. 감성 지수는 아래와 같은 방식으로 산출된다.

$$Sentiment Index = \frac{\sum_a Positive - \sum_a Negative}{\sum_a Positive + \sum_a Negative}$$



초록 그래프는 월별 감성 지수이고, 빨강 그래프는 코스피 월별 평균 지수이다. 감성 지수가 코스피 지수를 선행하는 모습을 볼 수 있었다.



감성 지수가 코스피 지수를 선행함을 정량적으로 분석하기 위해 감성 지수를 0~6 개월 이동하면서 상관계수를 산출하였다. 원자료의 상관계수는 0.31로 나타났으며, 감성 지수를 5 개월 이동하였을 때 상관계수가 0.6으로 나타나 감성 지수가 코스피 지수를 선행함을 정량적으로 분석할 수 있었다.

2.3 Granger Causality with KOSPI

그레인저 인과검정은 두 시계열 변수 사이의 인과 관계를 확인하는 통계적 방법이다. "인과 관계"란 "한 변수의 과거 값이 다른 변수의 현재나 미래 값을 예측하는 데 도움이 되는지"를 의미한다. 이 검정은 두 변수 중 어느 하나가 다른 하나의 원인이라고 주장할 수 있는지를 평가하는 데 사용된다.

우리는 그레인저 인과검정을 실시하여 시차를 둔 데이터가 코스피 예측에 도움이 되는지 검증하였다. [표 1]은 KOSPI 지수에 대해 최대 시차 12를 설정하여 그레인저 인과검정을 실시한 결과이다.

유의수준 5%에서 검정하였으며 감성지수는 모든 시차에서 유의함을 확인할 수 있었다. 벤치마크로 추가 선행한다고 흔히 알려져 있는 경기선행지수, M2, BTC 등과 비교해 보았을 때 확연히 유의함을 확인할 수 있었다.

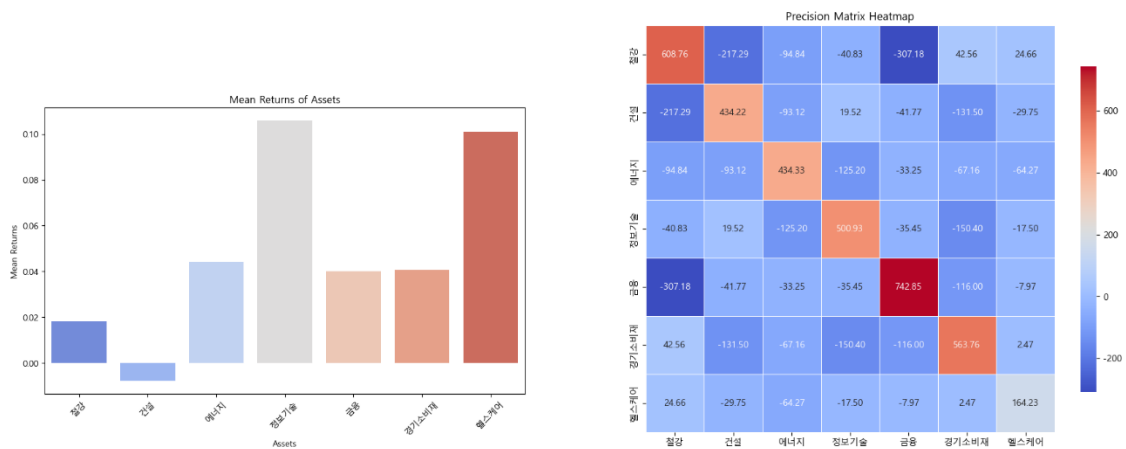
시차	경기선행지수	M2	BTC	sent-index
1	0.5699	0.707	0.6506	0*
2	0.0383*	0.8049	0.347	0*
3	0.0678	0.4673	0.5562	0*
4	0.2748	0.1211	0.5762	0.0001*
5	0.3913	0.0204*	0.4863	0.0001*
6	0.0609	0.0423*	0.5569	0*
7	0.0199*	0.0591	0.5448	0*
8	0.0536	0.0117*	0.5725	0.0005*
9	0.0469*	0.0066*	0.6216	0.0008*
10	0.0515	0.0136*	0.7625	0.0001*
11	0.0549	0.0106*	0.5006	0*
12	0.0597	0.0062*	0.1439	0*

[표 1] 그레인저 인과검정(Y = KOSPI, max lag = 12)

Chapter 3 : 연구 및 분석

3.1 Basic Mean-Variance Optimization

MVO 를 계산하기 위해 2011 1 월 ~ 2024 년 6 월까지의 KRX 건설, 철강, 에너지, 정보기술, 금융, 경기소비재, 헬스케어 index 데이터를 뽑아내었다. 일단 먼저 MVO 기본 weight 를 뽑아내기 위해 수익률이란 precision 행렬(cov matrix 의 역행렬)을 뽑아내었다.



그 결과 금융과 철강이 변동성이 크게 나타났으며, 정보기술과 헬스케어의 수익률이 가장 좋게 나타났다 위 데이터를 사용하여 무위험 자산 수익률을 3.5%로 잡고 앞서 설명한 공식대로 mvo weight 를 계산한 결과 다음과 같은 weight 가 추출되었다.

[2.82105022e-09, 7.35392569e-09, 5.44350608e-09, 7.86609304e-01, 9.20433771e-10, 1.07676183e-09, 2.13390673e-01, 5.64405253e-09]

앞서 설명한바와 같이 특정 자산에 자산배분이 과도하게 몰린 코너해 문제가 발생한 것을 알 수 있다. 이러한 mvo 계산을 바탕으로 매 달마다 백테스트를 돌렸을 경우 수익률은 다음과 같이 나온다.



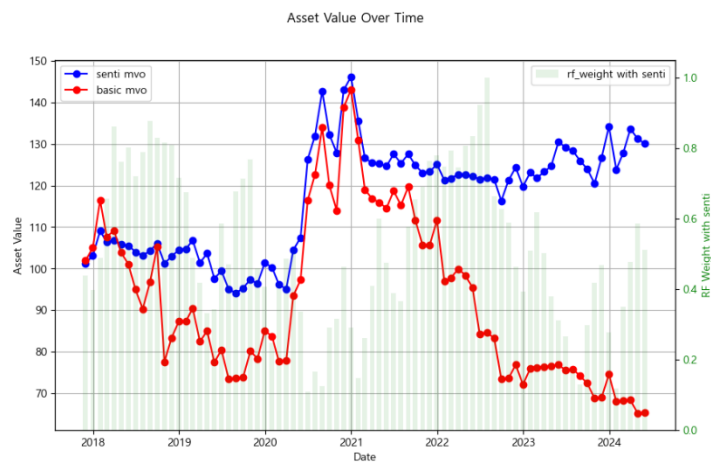
수익률이 좋지 않게 보이는 것을 관찰할 수 있다. 이는 mvo의 고질적인 문제인 코너해 문제가 발생해 무위험 자산 비중이 적어 위험에 과도하게 노출되어 있음을 나타낸다. 이를 해결하기 위해 우리는 sentimental data를 사용해 최적화하기로 결정했다.

3.2 sentimental data 뽑는 방식

앞서 설명한 방식대로 정규화된 sentimental score를 사용해서 mvo의 weight를 최적화해 보았다. 기본적인 개념은 다음과 같다.センチメンタル 지표와 코스피간의 상관관계 및 인과관계를 관찰해본 결과, 한달전의 senti 뿐만 아니라 두~세달 전의 senti 또한 코스피에 영향을 끼칠 수 있음을 알게 되었다. 이를 바탕으로 이전 senti 값들에 적절한 가중치를 곱하여 무위험 자산에 배분을하기로 결정하였다.

1. 무위험 자산 없이 mvo weight를 뽑는다.
2. $(1 - \text{한달전 sentiscore} * 0.4 + \text{두달전 senti score} * 0.3 + \text{세달전 senti score} * 0.3)$ 한 만큼의 weight를 무위험 자산에 투자한다.
3. 한달마다 리밸런싱을 진행하며 최종 수익률을 계산한다.

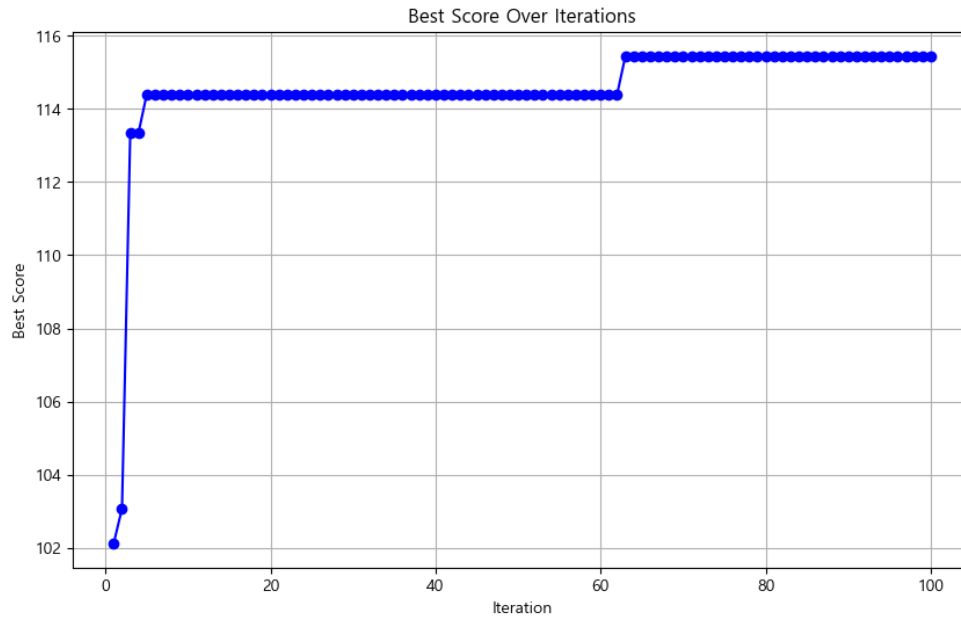
위 방식대로 진행한 결과 기본 mvo의 자산 배분과 sentimental data를 활용한 mvo 자산배분의 실험 결과는 다음과 같이 나왔다. 수익률이 급격하게 감소하는 경우 adj.senti score가 급격히 낮아져 rf에 투자하는 비중이 높아지고 수익률이 감소하는 경우에도 벤치마크 대비 큰 하락을 방지하는 효과가 있음을 보였다. 또한, basic mvo가 sentimental을 활용한 MVO와 기본 MVO의 경우 Sharp ratio와 MDD 면에서도 큰 차이를 보였다.



3.3 몬테카를로를 활용한 최적 weight 판단

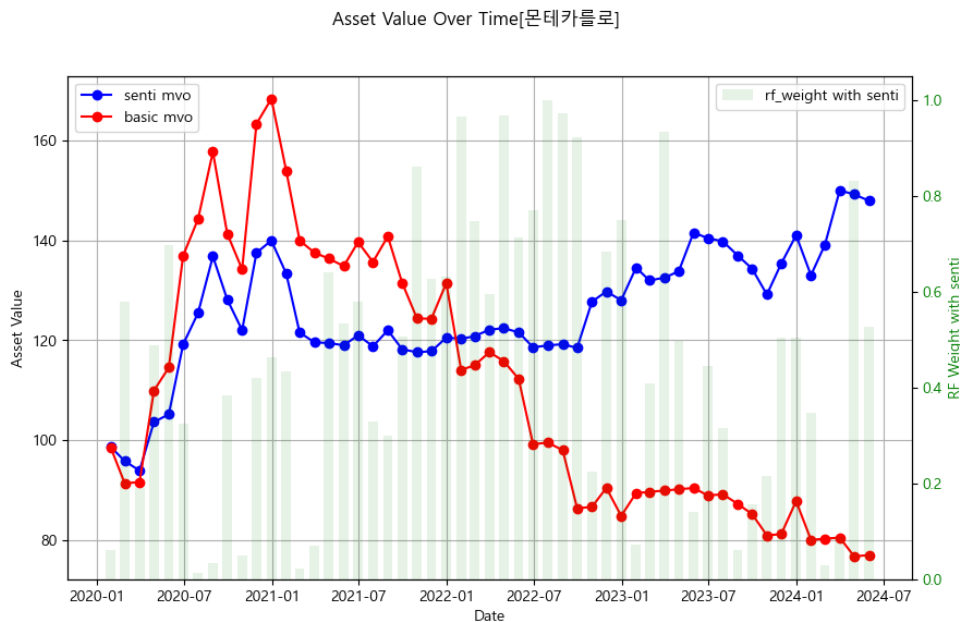
그 이후 weight를 변경시켜나가 보며 어떤 weight가 가장 백테스트에서 가장 최고의 결과를 나타내는지 관찰해 보았다. weight 계산은 몬테카를로적 방식을 사용하였으며 train data의 경우 2017~2019년 사이의 데이터를

사용하였으며 test data 의 경우 2020~ 2024 년까지의 데이터를 사용하여 결과를 관찰해 보았다. 총 100 번의 시뮬레이션 결과 수렴 곡선의 경우 다음과 같이 나타났다.

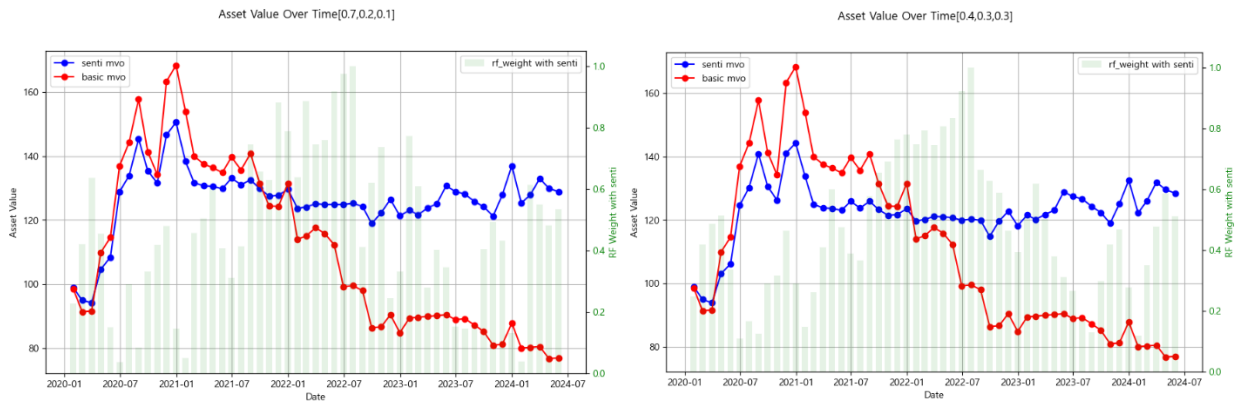


특정 값 이상부터는 증가하지 않아 100 번의 시뮬레이션 이후 종료하였다. 그 결과 best weight 는 다음과 같이 나왔다. [0.03326728958617764, 0.07934975858828042, 0.8873829518255419]

이후 weight 를 바탕으로 백테스트 결과 다음과 같이 나왔다. 파란색 선이 sentimental data 를 활용한 mvo 백테스트 결과이며 빨간색 선이 기본 mvo 를 활용한 백테스트 결과이다.



기본 벤치마크와 비교해본 결과는 다음과 같다.



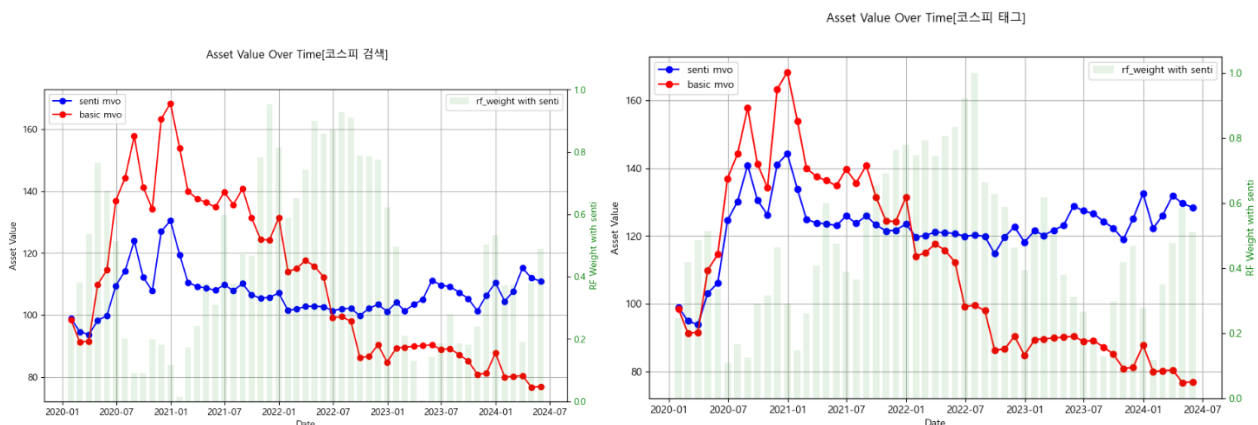
sentimental 을 통한 mvo 최적 예측은 3 개월전 sentimental data 에 가장 높은 비중을 두는 것이 훨씬 더 유리하다는 사실을 알 수 있었다.

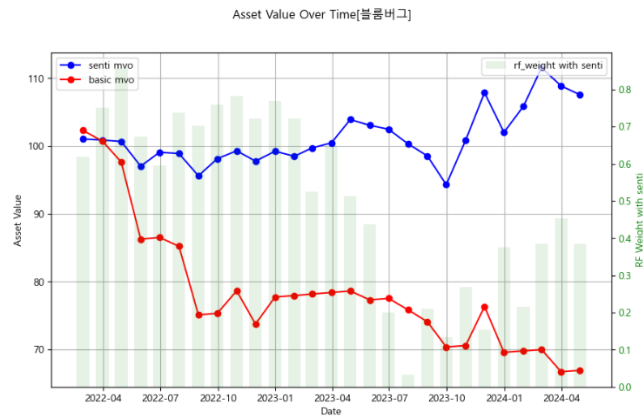
	Senti1 (M.C)	Senti2 (0.7,0.2,0.1)	Senti3 (0.4,0.3,0.3)	Basic Mvo
Sharpe Ratio	0.6512	0.4301	0.4354	-0.1245
MDD	-0.1599	-0.2046	-0.2104	-0.5445
Sortino Ratio	1.3679	0.8782	0.8812	-0.2293
Information Ratio	0.9012	0.8278	0.7919	Nan

3.4 Sentimental 데이터 변경

그 후 어떤 뉴스 데이터가 mvo weight 최적화에 가장 잘 맞는지 확인하기 위해 데이터를 바꿔서 계산해 보았다. weight 는 0.4,0.3,0.3 으로 고정하였다. 해외 매크로 데이터 같은 경우 가지고 있는 기간이 짧다보니 같은 길이를 통해 비교하기 힘들었다.

총 3 가지 데이터를 통해 관찰해본 결과 해외 매크로데이터 혹은 코스피 검색 데이터 보다 코스피 태그 데이터가 가장 뛰어난 성능을 나타내는 것을 관측할 수 있었다. 자세한 결과는 appendix.A 에 정리해 놓았다.





	Senti(코스피 검색)	Senti(코스피 태그)	Senti(해외 매크로)	Basic MVO
Sharpe Ratio	0.3525	0.3525	0.3567	-0.1245
MDD	-0.2046	-0.2351	-0.0924	-0.3479
Sortino Ratio	0.4162	0.8812	0.6840	-0.2293
Informatio Ratio	0.4654	0.7919	2.0457	Nan

Chapter 4 : 결과

본 연구에서는 KR-finBERT 모델을 활용하여 코스피와 관련된 감성 점수(sentiment score)를 산출하였다. 분석 결과, 감성 점수는 실제 코스피 200 지수에 선행하는 경향이 있음을 확인할 수 있었다. 감성 점수와 코스피 200 지수 간의 상관계수(Correlation)는 최대 0.6 으로 나타났으며, 약 5 개월 전의 감성 점수가 코스피 지수를 가장 잘 예측하는 것으로 분석되었다.

추가적으로, 그랜저 인과 검정(Granger Causality Test)을 통해 코스피와 다른 지표들 간의 인과 관계를 비교하였다. 그 결과, M2, 경기선행지수, 비트코인 가격 등 기존에 알려진 주가 선행 지표들보다 감성 점수가 주가를 예측하는 데 더 유의미한 것으로 나타났다. 이는 감성 점수가 기존의 선행 지표들보다 주가 예측에 더 효과적임을 시사한다.

이를 바탕으로, 감성 점수를 활용하여 MVO(Markowitz's Mean-Variance Optimization) 비중을 최적화하였다. 그 결과, 샤프 비율(Sharpe Ratio)이 기본 MVO 대비 -0.12 에서 0.43 으로 크게 개선되었으며, 최대 낙폭(MDD, Maximum Drawdown)은 -0.54 에서 -0.2 로 크게 감소하였다.

추가로 데이터셋을 코스피 태그 데이터, 코스피 검색 데이터, 해외 매크로 데이터로 비교 분석한 결과, 코스피 지수 예측에는 코스피 개별 기업이나 해외 매크로 지수보다 코스피 태그 데이터, 즉 코스피 자체 외국인 수급과 같은 데이터의 감성 지표가 더 우수한 성능을 보임을 밝혀내었다. 이는 코스피 태그 데이터가 다른 데이터에서 받은 감성 지표보다 코스피 지수 예측에 더 효과적임을 의미한다.

Reference

- 뉴스 감성 분석을 반영한 포트폴리오 선정 모형의 투자 성과 분석: 한국 주식시장을 중심으로 (박주미 외 3 인, 연세대학교 경영대학 경영학과)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. arXiv preprint arXiv:1908.10063.
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
- Banholzer, N., Heiden, S., & Schneller, D. (2019). Exploiting investor sentiment for portfolio optimization. Business Research, 12, 671-702.
- Markowitz, H. (1952). Portfolio Selection. The Journal of Finance, 7(1), 77-91.
<https://doi.org/10.2307/2975974>
- Fabozzi, F. J., Kolm, P. N., Pachamanova, D. A., & Focardi, S. M. (2007). Robust portfolio optimization and management. John Wiley & Sons.
- Li, X., Xie, H., Chen, L., Wang, J., & Deng, X. (2014). News impact on stock price return via sentiment analysis. Knowledge-Based Systems, 69, 14-23.
- Kalyani, J., Bharathi, P., & Jyothi, P. (2016). Stock trend prediction using news sentiment analysis. arXiv preprint arXiv:1607.01958.

Appendix A. 2022~ 2024 sentimental review

date	revenue1	rf_weight1	sent_score1	norm_sent12	revenue2	rf_weight12	sent_score2	norm_sent12	revenue3	rf_weight13	sent_score3	norm_sent13
2022-01-01 00:00:00			-1	0.827586207	-3.11612044	0.747932273	-1.944444444	0	-5.254133754	0.589574691	-0.306930693	0.011047037
2022-02-01 00:00:00	0.635965	0.29900083	-2.611111111	0	0.390336788	0.792544808	-0.238095238	0.30068985	0.45982866	0.652914008	0.04	0.123840946
2022-03-01 00:00:00	1.053663	0.618719212	-2.714285714	0	0.797990398	0.745886456	-0.479452055	0.258158377	0.802034241	0.743875115	0.123853211	0.151103254
2022-04-01 00:00:00	-0.16906	0.751724138	-2	0.358565737	-0.068365366	0.806529694	-2.425925926	0	0.101696206	0.899092304	0.125	0.151476097
2022-05-01 00:00:00	-0.19294	0.856573705	-1.619047619	0.549800797	-0.274027558	0.832345532	-3.357142857	0	-0.191758959	0.856926301	-0.051724138	0.094019647
2022-06-01 00:00:00	-3.62984	0.67250996	-2.058823529	0.329036794	-0.639217808	0.922552487	-3.5	0	-1.248413547	0.871618336	-0.87109375	0
2022-07-01 00:00:00	2.11272	0.595875322	-3.25	0	0.287089872	1	1.267857143	0.841386555	0.619797371	0.926351277	-0.457013575	0.156634626
2022-08-01 00:00:00	-0.18273	0.736348723	-2	0.495495495	-0.312637422	0.663445378	-1.777777778	0.303921569	0.125181328	0.909140255	0.204545455	0.411446895
2022-09-01 00:00:00	-3.32218	0.703090764	-2.666666667	0.231231231	-4.239119663	0.626015406	-2.535714286	0.170168067	-2.284774544	0.788430854	-0.585253456	0.109337872
2022-10-01 00:00:00	2.630216	0.758858859	-3.388888889	0	4.287113762	0.588340336	1.954545455	0.987357014	2.368040746	0.785840395	-0.423809524	0.171092412
2022-11-01 00:00:00	1.194794	0.781981982	-2	0.521821632	2.523562281	0.462830304	0.092783505	0.650349332	1.222499326	0.775327605	1.057777778	0.737820078
2022-12-01 00:00:00	-1.53446	0.721901978	-2.944444444	0.186046512	-3.691384261	0.392602743	-3.709677419	0	-2.197058038	0.620742884	0.636690647	0.576748418
2023-01-01 00:00:00	1.486034	0.769034906	-3	0.162790698	2.837499322	0.508688096	-1.0625	0.467350505	2.900109294	0.496626886	1.037453184	0.989462961
2023-02-01 00:00:00	-0.74091	0.722523278	-1.75	0.926008969	-1.128321612	0.617954998	1.366972477	0.896265915	-2.640302174	0.209844267	1.259259259	1
2023-03-01 00:00:00	1.241352	0.52494525	-3.272727273	0.065633918	1.28887248	0.501288483	-0.704918033	0.53048043	2.034420761	0.130136586	1.411067194	1
2023-04-01 00:00:00	0.753598	0.647106533	-2.55	0.473991031	1.108388005	0.378722902	0	0.6549314	1.604862442	0.003161112	0.600877193	0.644989981
2023-05-01 00:00:00	3.420297	0.512910722	-1.352941176	1	4.72549986	0.310003536	1.673469388	0.950376941	5.806158711	0.142004008	0.960199005	0.80243804
2023-06-01 00:00:00	-0.80933	0.438112515	-1.571428571	0.892685164	-1.148641069	0.264225674	1.15	0.857960135	-1.302205802	0.18552779	0.768518519	0.718447256
2023-07-01 00:00:00	-0.56662	0.200728625	-1.058823529	1	-0.593860598	0.175223444	0.953488372	0.82326665	-0.483664696	0.278392691	1.200956938	0.894751249
2023-08-01 00:00:00	-2.09393	0.032194451	-2.095238095	0.555199359	-1.857755972	0.128192217	-0.911764706	0.493962327	-1.715944266	0.185833912	1.086206897	0.837270482
2023-09-01 00:00:00	-1.79108	0.210114707	-0.764705882	1	-1.559732622	0.298047034	-1.066666667	0.466614893	-1.867332778	0.181132256	0.614906832	0.601186131
2023-10-01 00:00:00	-4.2982	0.133440192	-1.65	0.6626401	-2.791509448	0.418185349	-0.261538462	0.608757642	-3.734783349	0.239919028	-0.119815668	0.165675357
2023-11-01 00:00:00	6.967791	0.268384152	-1.090909091	0.869936034	5.14205552	0.468323777	3.255813953	1	4.83278215	0.502192873	0.790780142	0.59481743
2023-12-01 00:00:00	6.943518	0.153233556	-2.235294118	0.413646055	5.967538127	0.27738824	5.876712329	1	3.965921439	0.532014581	1.657608696	1
2024-01-01 00:00:00	-5.42726	0.374768738	-0.733333333	1	-7.779746555	0.117372708	-0.194029851	0.125678984	-5.45391303	0.371852164	0.529880478	0.365526747
2024-02-01 00:00:00	3.759612	0.214925373	-2.055555556	0.479315831	3.163354715	0.349728407	2.15625	0.464171215	3.050052153	0.375344072	1.914965996	1
2024-03-01 00:00:00	5.390375	0.384179851	-2.619047619	0.257415615	4.624260742	0.476627819	3.040322581	0.591497202	6.996703648	0.190341976	0.943089431	0.522268136
2024-04-01 00:00:00	-2.41712	0.453329004	-0.761904762	0.984848485	-1.758296334	0.58644606	1.924242424	0.430756998	-2.772454764	0.381394722	0.15498155	0.135049978
2024-05-01 00:00:00	-1.18186	0.385041172	-1.15	0.779040404	-0.880993824	0.510996676	6.236842105	1	-0.93289343	0.489269568	1.771028037	0.929261231