지역 투자 정책을 이용한 강화학습 기반 동적 자산 할당 기법

(A Dynamic Asset Allocation Method based on Reinforcement Learning Exploiting Local Traders)

오 장 민 † 이 종 우 †† 장 병 탁 †††
(Jangmin O) (Jongwoo Lee) (Byoung-Tak Zhang)

요 약 본 논문에서는 패턴 기반의 다수의 주가 예측 모델에 기반한 지역 투자자의 효율적인 결합을 통해, 거래 성능을 최대화 할 수 있는 동적 자산 할당 기법을 연구하였다. 각 예측 모델이 추천한 후보 종목에 효과적인 거래 대금 비율을 할당하는 메타 정책(meta policy)이라는 자산 할당 정책을 강화 학습 틀 내에서 정의하였다. 이를 위해 각 예측 모델의 추천 종목 수와 전체 자산 대비 주식 자금 비율을 동시에 활용하는 상태 공간을 설계하였다. 대한민국 주식 시장에 대한 시뮬레이션 실험을 통해, 본 논문에서 제안한 자산 할당 정책은 기존의 고정 자산 할당 방법들에 비해 우수한 성능을 보임을 제시 하였다. 이는 강화학습을 통한 지역 투자자의 결합을 통해 의사 결정 문제에서 감독자 학습 기법으로 학습된 예측 모델의 시너지 효과를 거둘 수 있음을 의미한다.

키워드 : 주식 거래, 자산 할당, 강화학습

Abstract Given the local traders with pattern-based multi-predictors of stock prices, we study a method of dynamic asset allocation to maximize the trading performance. To optimize the proportion of asset allocated to each recommendation of the predictors, we design an asset allocation strategy called meta policy in the reinforcement learning framework. We utilize both the information of each predictor's recommendations and the ratio of the stock fund over the total asset to efficiently describe the state space. The experimental results on Korean stock market show that the trading system with the proposed meta policy outperforms other systems with fixed asset allocation methods. This means that reinforcement learning can bring synergy effects to the decision making problem through exploiting supervised-learned predictors.

Key words : stock trading, asset allocation, reinforcement learning

1. 서 론

최근 수십년 동안, 주식 시장의 문제들을 접근하는데, 여러 기법들이 활용되어 왔다[1]. 그러나 주식 시장을 모델링 하거나 예측하는 시도들은 항구적으로(consistently) 시장을 극복하는데 성공적이지 못했다. 이것이 유명한 효율적 시장 가설(Efficient Market Hypothesis) 로서, 시장의 모든 가용 정보는 현재까지의 시장 상태에 반영이 되어 있기 때문에, 미래의 주가는 예측 불가하다는 것을 의미한다[1-3]. 그러나, 항구적인 조건을 약간 완화 시킨다면, 최근의 경향은 시장이 완벽하게 예측 가능하지는 않더라도 어느 정도 이익을 유도할만 하다는 것이다[3]. 특별히, 인공지능 분야의 최신 알고리즘들은 강력한 표현력과 모델링 능력을 지니고 있으며, 이들을 이용한 가격 예측, 위험 조절, 포트폴리오 최적화 응용 예들이 보고되고 있다[4].

가격 예측 기법에는 문제의 기술면에서 감독자 학습 기법이 잘 들어맞으며, 신경망(Artificial Neural Networks), 결정 트리(Decision Trees), SVM(Support Vector Machines) 등이 적용 되어 왔다[5-8]. 위험 관리와 포트폴리오 최적화는 강화 학습 기법 내에서 집중

· 이 논문은 교육인적사업부의 BK21 사업과 과학기술부의 국가지정연구실 사업(NRL)과 산업자원부에 의해 지원되었음.
† 학생회원 : 서울대학교 컴퓨터공학부
jmoh@bi.snu.ac.kr
†† 종신회원 : 숙명여자대학교 멀티미디어학과 교수
bigrain@sookmyung.ac.kr
††† 종신회원 : 서울대학교 컴퓨터공학부 교수
btzhang@bi.snu.ac.kr
논문접수 : 2005년 3월 2일
심사완료 : 2005년 6월 18일

적으로 취급되어 왔다[9-12].

우리는 그동안 개별 주식 투자를 위한 기계 학습 기법 연구에 집중해왔다[7,9,12]. 그러나 개별 주식 투자 문제에서 그동안의 감독자 학습 기법이나 강화학습은 한계를 지니고 있다.

감독자 학습에 기반한 주가 예측 기법은 위험 관리나 포트폴리오 최적화를 고려하지 못하거나 그러기에 적합하지 않다[6-8]. 입력과 지정된 목표값과의 정적인 매핑 관계를 학습하는데 적합한 반면, 동적인 결정 과정을 학습 과정에 포함시키기가 곤란하다. 이들 감독자 학습 기반 방법들은 근본적으로 자산 할당 기법을 통합하지 못하고 있다. 강화학습의 축적된 보상을 이용한 시지연 학습 메카니즘은 위험 관리나 포트폴리오 최적화 등의 의사 결정 문제에 적합하다. 그러나, [11]의 연구는 강화학습 틀 내에서 문제가 취급 가능하도록 무리한 가정을 두었다. 또, [10]에서의 포트폴리오는 간단하여, 오직 두 개의 가격 사이에서 자산을 전환하는 것만 가능했다. [9, 12]의 연구에서는 강화학습 내에서 개별 주식을 거래 가능하도록 전개 했지만, 자산 할당을 고려치 못했다. 최근에는 감독자 학습 방법을 강화학습 으로 보완하여 금융 문제에 적용하는 연구가 태동하고 있다[13].

본 논문의 연구 목표는 주식 선택과 자산 할당 정책이 통합된 주식 거래 환경의 구현에 있다. 전통적인 감독자 학습과 강화 학습은 이 둘을 동시에 취급하기에 어려움이 있었다. 본 논문에서는 문제를 분할 후 통합하는 방법을 택하였다. 즉, 자산 할당과 분리하여, 주식 예측 및 선택을 담당하는 지역 투자자를 감독자 학습으로 최적화를 먼저 행하였다. 자산 할당 정책은 생성된 지역 투자자를 통합하여 자산을 최대화 할 수 있도록 강화학습을 통해 최적화 한다. 본 논문은 이미 구성이 완료된 지역 투자자를 의사 결정 차원에서 더 활용하여 효과적인 자산 할당 정책을 구성할 수 있는 강화학습 전개에 초점을 맞춘다.

논문의 구성은 다음과 같다. 2장에서는 구성 완료된 지역 투자자에 대해 간단히 요약하고, 3장에서는 지역 투자자를 이용하는 효과적인 자산 할당 정책인 메타 정책에 대하여 기술한다. 4장에서는 메타 정책을 최적화의 강화학습 전개에 대해 설명한다. 5장에서는 실험을 통한 메타 정책의 성능 및 결과를 분석하고, 6장에서는 토론을 그리고 7장에서 결론을 맺는다.

2. 지역 투자자의 요약

이 장에서는 4장에서 기술할 주식 투자 모형의 기본 구성 요소인 지역 투자자의 구성에 대해서 간단히 요약한다. 이를 위해 그림 1에 지역 투자자(Local Trader)의 개념을 다시 표현하였다. 지역 투자자 LT_e는 특정

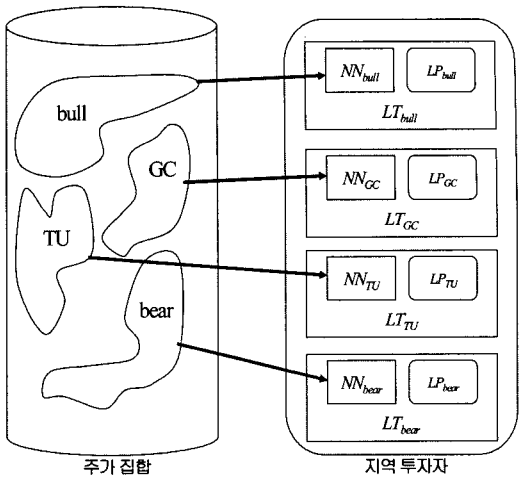

그림 1 지역 투자자 개념. 전체 주식 데이타 중 경험상 의미 있는 패턴 4가지를 선정한 후, 패턴 e이에 대응되는 신경망 기반 예측 모델 NN_e와 지역 정책 LP_e로 구성된 지역 투자자 LT_e를 구성하였다.

주가 패턴 e에 대한 예측 모델 NN_e와 지역 정책 LP_e로 구성된다. 주식 시장은 불확실성이 큰 문제 중 하나이기 때문에 우리는 모든 주가 패턴을 공략하지 않고, 의미 있는 몇 개의 패턴에 집중하였다. 표 1은 우리가 집중 연구한 패턴에 대한 설명이다. 주가 데이타에 대한 우리의 접근법은 그랜빌의 이동평균선간 배열에 따른 특성들에 기반한 기계학습 기법의 적용이다[1,9]. D_{bull}은 상승형 패턴 일부를 말하는데, 5, 10, 20, 60, 120일 이동평균선들이 모두 순서대로 다른 선들보다 위에 위치하는 정배열의 패턴을 의미하며, D_{bear}는 반대로 모두 역배열인 것을 의미한다. D_{GC}는 단기선들과 장기선들간에 단기선이 장기선을 상승 관통하는 골든 크로스가 특정 기간내에 발생한 패턴을 의미한다. D_{TU}는 어떤 이동평균선의 방향이 하락에서 상승으로 바뀐 사건이 특정 기간내에 발생한 패턴을 의미한다.[1] 지역 투자자 생성에 사용[2]한 대한민국 거래소 데이타 중 19.7%, 20.3%, 9.7%, 12%가 각각 D_{bull}, D_{bear}, D_{GC}, D_{TU}에 해당한다. 각 지역 투자자는 2 단계 과정을 거쳐 완성 되었다.

첫 번째는 예측 모델의 최적화 단계로서 각 주가 패턴 별로, NN_{bull}, NN_{bear}, NN_{GC}, NN_{TU}의 신경망 기반

1) 패턴들에 대한 보다 자세한 설명은[1]을 참고 바란다.

2) 대한민국 거래소 시장에 속한 주식을 대상으로 하며, 학습기간은 1998년 1월부터 2000년 3월로 삼았다. 이중 후반부 1/3은 검증기간으로 삼았다.

표 1 본 논문에서 사용된 주식 데이타들의 패턴 정보

패턴	예측모델	의미
D_{bull}	NNbull	이동평균선들이 정배열 상태
D_{bear}	NNbear	이동평균선들이 역배열 상태
D_{GC}	NNGC	이동평균선들간 골든크로스가 발생
D_{TU}	NNTU	이동평균선의 하락에서 상승 반전이 발생

예측 모델을 구성하였다. 우선 학습 기간에 대해 각 패턴에 해당하는 학습 데이타와 검증 데이타 T_e, V_e를 모은다.

$$T_e = \{S \mid S \in D_{e,t}, t = tday_1, ..., tday_{|T|}\},$$

$$V_e = \{S \mid S \in D_{e,t}, t = vday_1, ..., vday_{|V|}\}$$

즉, 전체 학습 기간의 날짜 수를 $|T|$라고 할 때, 모든 날짜 t에 대해, 패턴 $D_{e,t}$에 해당하는 주가 데이타 S를 모아서 예측 모델 e에 대한 학습 데이타 T_e를 구성한다. 각 S는 입력 S_{IN}, 타겟 S_{OUT} 쌍 으로 구성된다. 패턴 D_e 에 대한 지역 투자자의 신경망 NN_e의 최적화는 다음과 같이 모델 가중치 벡터 $\mathbf{w}_e$를 찾는 식으로 나타낼 수 있다.

$$\arg\min_{\mathbf{w}_e} \frac{1}{2} \sum_{S \in T_e} \{NN_e(s_{IN}; \mathbf{w}_e) - s_{OUT}\}^2 \quad (1)$$

즉, 학습 데이타 T_e에 대해, 에러 제곱합을 최소화하는 $\mathbf{w}_e$를 찾는 것이다. 신경망은 활성화 함수로 초탄젠트(hypertangent) 함수를 사용하는 뉴런으로 구성된 은닉층이 2개인 구조로 제한했으며,[3] 가중치 갱신은 SCG (scaled conjugate gradient) 방법을 사용하였다. 이 때, 학습은 매 갱신 마다 검증 데이타 V_e에 대한 에러 제곱합의 경향을 관찰하여 중단 하게 된다. 또한 은닉층의 뉴런 수를 변화 시켜가며 패턴 e 마다 최적의 $\mathbf{w}_e$를 찾는 절차를 택하였다.

두 번째 단계는 지역 정책의 최적화 단계이다. 주가 시계열은 특성상 많은 잡음을 내재하기 때문에 에러를 최소화 하는 것과 별도로 예측 모델의 활용에 따라 투자 성능이 크게 변할 수 있다. 즉, 예측치가 어느 범위에 있을 때 매수 및 매도를 해야 하는지, 또 어느 정도 기간 동안 보유를 해야 하는지의 결정에 따라 투자 성능이 상이할 수 있다. 이를 담당하는 지역 정책을 표 2에 나타내었다. 지역 정책은 예측 모델을 활용하여 다음과 같은 매수 신호, 매도 신호의 결정을 담당한다.

매수 신호: 종목 S에 대한 예측치 score(S)가 bid_thres 보다 클 때

표 2 지역 정책의 파라미터

이름	의미
bid_thres	매수 신호의 임계치
ask_thres	매도 신호의 임계치
hold_due	주식 보유의 만료일

매도 신호: 보유 중인 종목(S')에 대한 예측치 score(S')가 ask_thres 보다 작거나, 보유기간이 만료일 hold_due 에 달했을 때

지역 정책의 최적화 수단으로 우리는 시뮬레이션 기법을 사용하였다. 지역 정책의 최적화는 지역 정책은 시뮬레이션은 다음과 같은 매수/매도 신호를 통해 이뤄진다.

```
N = 0; L = 전체 주식 후보의 수;
for i = 1 : L
    만일 i번째 후보에 대해 매수 신호 발생하면
        매도 신호에 따른 매도;
        profit_i =(ask_i - bid_i - tr_i) / bid_i;
        N = N+1;
end for
PPT = (1/N) Σ profit_i 리턴;
```

여기에서 N은 전체 거래 수이다. 또, ask_i는 매도가, bid_i는 매수가, tr_i는 거래세 대금이다.[4] 시뮬레이션은 예측 모델 최적화시 검증기간으로 확보 되었던 집합에 대하여 행하였으며 $profit_i$는 거래세가 반영된 이익률이다. PPT(Profit ratio per Trade)는 시뮬레이션의 점수로 거래당 평균 이익률을 의미한다. 시뮬레이션을 반복하여 최적의 파라미터를 찾는 것은 다음과 같이 수식으로 표현 할 수 있다.

$$\underset{\{ask\_thres\},\{bid\_thres\},\{hold\_due\}}{\text{argmax}} \quad PPT(ask\_thres, bid\_thres, hold\_due) \quad (2)$$

여기에서, {ask_thres}, {bid_thres}, {hold_열}는 파라미터의 가능한 집합이다. 예측 모델의 최적화는 입력과 타겟 사이의 에러를 최소화 하는 식 (1)의 감독자 학습인 반면, 지역 정책 최적화 식 (2)는 PPT를 최대화 시키는 지역 정책 파라미터를 찾는 것이지만 타겟을 미리 알 수 없고, 또한 미분 가능하지도 않다. 따라서, 예측 성능의 최적화는 감독자 학습으로, 지역 정책 최적화는 시뮬레이션을 통한 분리된 방법을 택하였다. bid_thres는 0.25~0.54까지 0.01씩 증가시켜가며 30개, ask_thres

3) 입력 차원은 138차원이며, 실수 피쳐 80, 이산화 피쳐값 58개로 구성되어 있다.

4) 거래세는 매도가 대비 0.8%를 고정사용하였다.

는 -0.54~-0.25까지 30개, hold_due는 5일부터 15일까지 10개로서, 세 파라미터의 조합은 9,900 경우의 탐색 공간으로 그리 크지 않기 때문에, 간단하게 무차별 대입법(brute-force)으로 최적의 파라미터를 찾는 전략을 취했다.

표 3은 이를 통해 최적화된 지역 투자자의 성능 결과를 나타낸다. 패턴별 분할 없이 전체 주가 데이타를 다루는 단일 지역 투자자 LT_{all}을 추가하였다. 두 번째 열은 정확률을 나타낸다. 이는 전체 거래 중 이익의 크기에 관계없이 이익을 산출한 거래의 비율이다. 세 번째 열은 PPT 이며, 네 번째 열은 최적화된 신경망의 구조로서, 첫 번째 은닉 층과 두 번째 은닉 층의 뉴런의 수를 기록하였다. 패턴에 따라 최적화된 신경망의 구조는 서로 다르다. 다섯 번째 열은 각 패턴에 해당하는 주가 데이타의 비율이다. 패턴 분할 접근에서 각 패턴마다 정확률은 69% 이상, PPT는 1% 이상의 값을 지닌다. 네 가지 패턴은 전체 주가 데이타 중 약 70% 정도 밖에 포함하지 못하지만, 이 접근법은 단일 지역 투자자에 비해 그 예측 및 투자 성능을 향상 시킬 수 있음을 의미한다.[5] 그러나 이는 자산 상황에 무관하게 추천 후보가 모두 매수 되었을 때의 이상적인 이익률을 의미한다. 실제 투자는 긴 투자 기간 동안에 한정된 자산을 활용해야 하므로 복잡한 자산 활용 정책까지 고려되어야 한다. 다음 장에서 이렇게 구성된 지역 투자자를 활용하는 효과적인 자산 활용 정책의 필요성에 대해 설명하겠다.

표 3 지역 투자자의 성능

지역 투자자	정확률(%)	PPT(%)	신경망구조	범위(%)
LTall	57.10	0.79	72x16	100.0
LTbear	69.42	1.41	60x15	20.3
LTbull	73.37	1.99	58x18	19.7
LTGC	71.10	1.72	25x14	9.7
LTTU	70.97	1.63	30x10	12.0

3. 메타 정책의 필요성

그림 2는 예측 모델 NN_{bear}와 이의 지역 정책을 이용한 지역 투자자 LT_{bear}의, 성공적 추천 및 실패한 추천을 바 그래프(Bar Graph) 형태로 나타낸 것이다. 가로축은 약 380일의 거래일이며, 세로축은 해당 거래일에 거래 완료된 주식들의 이익률(profit$_i$)의 합이다. 이 그림은 LT_{bear}의 추천된 종목들이 모두 거래 되었다고 가정했을 때의 결과이다. 하락 패턴인 D_{bear}에 기반한 예측

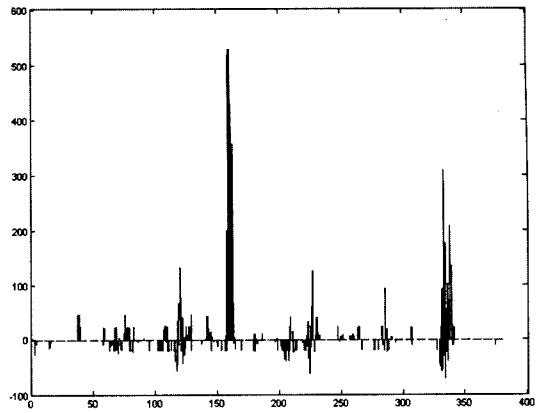

그림 2 지역 투자자 LT_{bear}의 추천 경향: 가로축은 거래일을 나타낸다. 세로축은 해당 거래일의 추천들의 이익률의 합을 나타낸다. 위쪽 바(bar)는 그날의 추천들이 이익을 냈음을, 아랫방향 바는 손실을 의미한다.

모델이 제출한 추천 중, 큰 이익을 유도하는 추천은 전체 거래 기간에 고르게 분포하지 않음을 알 수 있다. 특정 기간에 집중되어 있는 거래가 전체 투자 성능을 좌우 하는 경향을 보인다.

실제 거래에서는 한정된 자산을 활용하여 거래가 이뤄지므로, 추천당 매수 금액 PMR(Purchase Money per Recommendation)을 달리하여 거래 시뮬레이션을 실시해보았다. 두가지 거래 대금 40만원과 4,000만원에 대한 실험 결과를 각각 그림 3, 4에 보였다. 초기 자산은 두 경우 2억원으로 출발하였다. 각 그림에서 윗부분은 전체 자산의 변화를 나타내고, 아랫부분은 바 그래프를 나타낸다.

작은 PMR을 이용하면, 160부근이나 340부근 거래일 같이 추천수가 폭발적으로 증가한 날에도 대부분의 추천된 주식을 거래하는데 문제가 없다. 큰 이익을 볼 수 있는 이러한 거래는 자산 증가에 큰 몫을 하지만, 추천수가 많지 않은 보통의 거래일에는 총 자산 대비 매우 작은 비중만이 주식 거래에 활용된다. 총 자산 대비 투자금 비율이 너무 작으므로, 투자의 결과가 자산의 변동에 영향을 거의 미치지 못하는 경우가 발생한다.

반대로, 그림 4처럼 큰 PMR의 경우에는, 개별 거래에 의해 총 자산이 너무 크게 흔들릴 수 있다. 또한 적은 수의 매수로도 전체 여유 자산이 고갈 될 수 있다. 만일 그림 2의 160, 340 부근의 폭발적인 추천 수를 보이는 날에 매수된 소수의 종목이 이익을 주지 못한다면, 다수의 이익 유발하는 종목을 놓치는 상황이 벌어질 수 있다. 즉, 추천의 이익률에는 편차가 존재하는데, 손실을 끼치는 추천에 자산이 의존되어 있을 경우, 자산에 큰

5) 논문에서 사용한 4가지 패턴외 추가 패턴을 고려할 수 있겠지만, 5% 이상을 차지하는 유의미한 패턴의 발견이 쉽지 않았다. 또한 강화 학습 틀에서 자산 할당 고려시, 지역 투자자의 증가에 따른 문제의 복잡도가 크게 증가하므로, 본 논문에서는 이 4가지 패턴에만 국한한다.

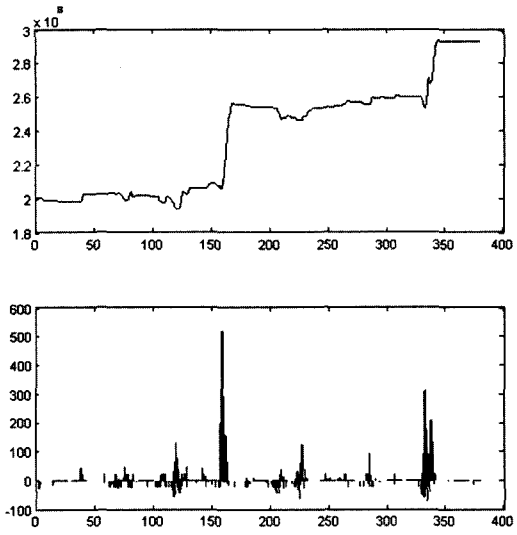

그림 3 추천당 매수 금액을 40만원인 경우. 윗그림은 초기 자산 2억원일 때 거래일에 따른 자산의 추이를 나타낸다. 아랫그림은, 모든 추천이 거래됐다는 가정하의 이익률(%)의 합을 나타낸다.

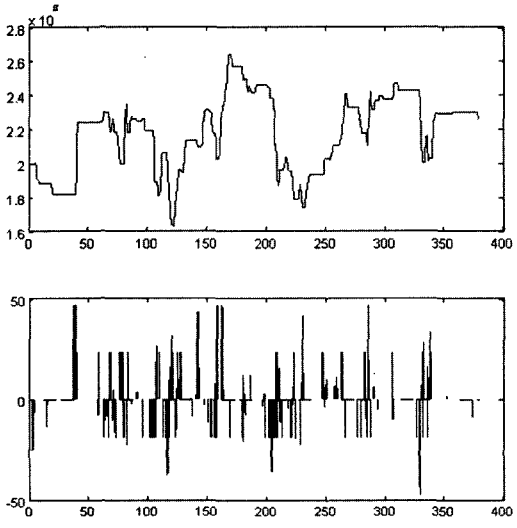

그림 4 추천당 매수 금액이 4,000만원인 경우. 윗그림은 초기 자산 2억원일 때 거래일에 따른 자산의 추이를 나타낸다. 아랫그림은, 모든 추천이 거래됐다는 가정하의 이익률(%)의 합을 나타낸다.

요동이 생길 수 있다.

그러므로 보다 효율적인 거래를 위해, 상황에 따라 PMR을 동적으로 변경해줄 수 있는 전략이 필요하다. 또한, 다수의 지역 투자자가 존재하므로, 각 예측 모델의 추천마다 PMR을 효과적으로 조절할 수 있는 복잡한 자산 할당 기법이 필요하다. 이를 위해, 우리는 메타 정책(Meta Policy)를 정의한다.

정의 1: 예측 모델의 추천 수 벡터를 N =($\#_{bear}$, $\#_{bull}$, $\#_{GC}$, $\#_{TU}$)라고 하자. 또, 총 자산 대비 주식 자금의 비율을 SF(stock fund ratio)라고 하자. 메타 정책 MP (meta policy)는 $\mathbb{R}^4\times\mathbb{R}\mapsto\mathbb{R}^4$ 인 함수로서,

$$(PMR_{bear}, PMR_{bull}, PMR_{GC}, PMR_{TU}) := MP(N, SF)$$

이다. 여기에서 PMR_e는 e번째 지역 투자자의 추천당 매수 금액(Purchase Money per Recommendation)이다.

그림 5는 메타 정책을 이용하는 거래 과정을 나타낸다. 이는 그림 6의 주식 거래 모형에서의 거래 절차로 생각할 수 있다. 전체 거래 기간 T 중 t번째 거래일에, E개의 예측 모델들은 각 패턴 D_e에 대한 추천 후보 집합 $\{S_e\}$를 검색한다. 메타 정책 MP는 각 예측 모델의 추천당 매수 금액을 결정하며 이는 그림 6의 자산 할당 정책에 해당한다. 이 때, PMR 결정은 예측 모델의 추천 종목 수 $\#_1$, ..., $\#_E$와 추천당 매수 금액 SF를 고려하여 계산된다. local_trade 함수는 해당 지역 투자자의 거래를 행한다.

주식 자금 비율과 추천 수에 대한 정보는 시간에 따라 변하고 의존적인 관계에 있으므로, 강화학습의 철학에 부합된다. 이에 우리는 대표적 강화학습인 Q-학습 하에서 메타 정책을 모델링하고자 한다.

```
for t = 1 to T
    for e = 1 to E
        ({S_e}, N_e)= retrieve(NN_e, LP_e)
    end for
    (PMR_1, ..., PMR_E) = MP(#_1,...,#_e, SF)
    for e = 1 to E
        local_trade(PMR_e, {S_e}, LP_e)
    end for
end for
```

그림 5 메타 정책에 따른 투자 과정. e 번째 지역 투자자에 대하여 retrieve 함수는 매수 신호가 발생한 주식 후보 집합 $\{S_e\}$를 검색하고, 후보의 수 N_e를 리턴한다. MP는 PMR을 정해주는 함수이다. local_trade는 매수된 종목에 대해 지역 투자 정책을 따라 거래 뒤처리를 행한다.

4. 강화 학습에 의한 자산 할당 모델

4.1 강화 학습

강화학습은 목표 지향적 학습 및 의사 결정 문제를 자동화하기 위한 계산학적 접근법이다[14]. 강화학습, 특

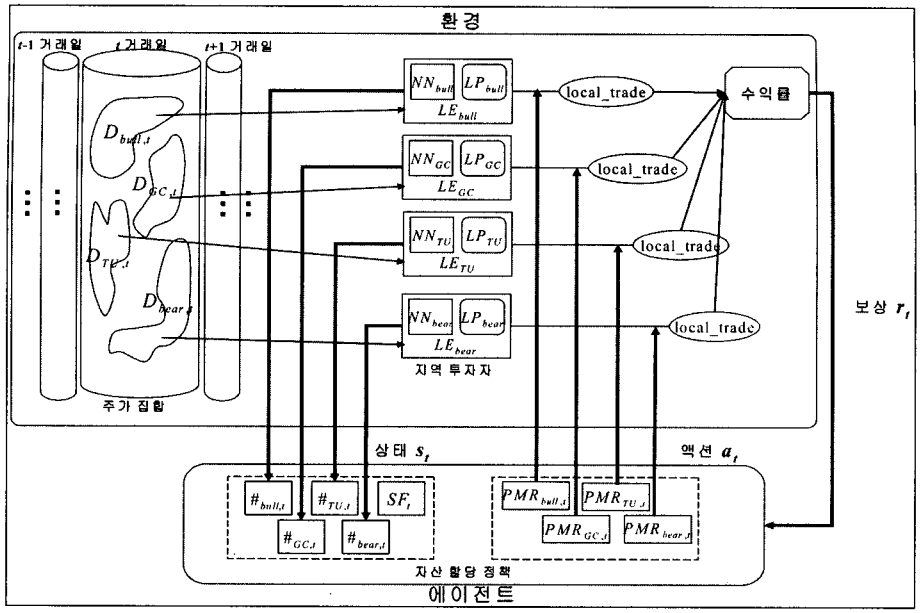

그림 6 강화학습으로 표현한 주식 거래 모형. 주식 시장은 환경, 자산 할당 정책은 에이전트에 해당한다. $\#_{e,t}$는 e번째 지역 투자자가 검색한 투자 후보의 수이다. SF_t는 총 자산 대비 주식 자금의 비율이다. $PMR_{e,t}$는 액션에 해당하며 e번째 지역 투자자의 추천당 매수 금액이다. local_trade는 지역 투자자의 매수, 매도를 행한다. 보상은 식 (3)의 수익률이다.

히 마코프 결정 과정(Markov Decision Process)에서는, 에이전트(agent)와 환경(environment)이 매 시간 t = 1,2,...,T 마다 상호작용한다. 환경의 상태 s_t에 대하여 에이전트는 자신의 정책 π으로부터 액션 a_t를 선택한다. 에이전트의 액션 a_t에 대응하여 환경은 상태를 s_{t+1}로 변경하고, 에이전트에게 액션에 대한 보상 r_{t+1}을 제공한다.

에이전트의 목표는 최적의 정책을 학습하는 것이며, 이는 상태-액션 쌍(s,a)에 대한 보상의 기대치를 최대화하는 것을 의미한다. Q-학습에서는 이 보상의 기대치를 함수로서 유지하며, s_0에서 출발하여 s_T에서 종료하는 주어진 에피소드에 대하여,

$$Q^{\pi}(s,a) = E_{\pi} r_t + \gamma r_{t+1} + \ldots + \gamma^{T-t-1} r_T | s_{T=s}, a_t = a$$

과 같이 정의된다. 여기에서 $0 \leq \gamma \leq 1$ 은 감쇄 요소(discount factor)이다. Q-학습에서 최적의 정책 Q^*는

$$Q^* = \arg\max_{\pi} Q^{\pi}(s,a)$$

과 같이 정의된다. 최적의 정책을 학습하는 방법으로는 다이나믹 프로그래밍과 몬테칼로 방법, 시간차(temporal-difference) 방법이 있다. 다이나믹 프로그래밍은 정확한 계산 및 수렴을 보장하지만, 환경에 대한 정확한 확률 모델이 존재하여야 하고, 몬테칼로 방법은 많은 수의 에피소드가 필요하다. 시간차 방법은 두 방법의 절충안으로서 n-단계 샘플과 부트스트랩 기법을 사용한다[14].

4.2 Q-학습을 이용한 자산 할당 모델

기계 학습 분야에서 바람직한 성능을 거두는 것은 입력 공간의 표현에 좌우된다. 특히, 강화 학습은 상태와 보상의 설계의 미학이다[12]. 이 절에서는, Q-학습 기법 내에서 메타 정책을 어떻게 설계했는지 자세히 기술하겠다.

그림 6은 강화 학습으로 표현한 전체 주식 거래 모형을 나타낸다. 상태 s_t는 환경이 에이전트에게 제공하는 시간 t에서의 상태 벡터이다. 우리는 s_t를 다음과 같이 설계하였다.

$$s_t = (N_{bear}^{bits}(t), N_{bull}^{bits}(t), N_{GC}^{bits}(t), N_{TU}^{bits}(t), SF^{bits}(t))$$

상태 벡터는 두 부분으로 나눌 수 있는데, 하나는 예측 모델에 대한 요약 정보이고, 다른 하나는 주식 자금 비율에 대한 정보이다. 예측 모델에 대한 정보는 $N_e^{bits}(t)$로서, 이는 시간 t에서의 e번째 지역 투자자의 추천 종목수 $\#_t^e$를 이산화된 값으로 표현한다. 추천 종목수는 상한에 제한이 없는 정수 값이므로, 표 4와 같은 이산화 방법을 사용하였다. 자연수 값을 10 비트의 직교 좌표로 표현 하였다.

$SF^{bits}(t)$는 총 자산 대비 주식 투자금의 비율을 표현하는 비트 벡터이다. 이 비율은 0%~100%의 값을 가질

표 4 예측 모델의 추천에 대한 비트 벡터 표현

추천 수	비트 벡터
0	0000000001
1	0000000010
...	...
8	0100000000
9~	1000000000

표 5 주식 자금 비율의 비트 벡터 표현법

주식대금의 비율	비트 벡터
[0,5)	00000000000000000001
[5,10)	00000000000000000010
...	...
[90,95)	01000000000000000000
[95,100)	10000000000000000000

수 있기 때문에, 이를 표 5와 같이 20 구간으로 나누고 20 비트의 직교 좌표로 표현했다.

상태 s_t에 대한 액션 a_t은 총자산에 대한 각 예측 모델의 PMR 비율(%)이다. 실수값의 PMR 할당이 자연스럽지만, Q-학습 기법 내에서 실수값의 액션을 처리하기는 쉽지 않다. 따라서 액션 또한 이산화 과정을 거쳐 적절한 차원의 벡터로 표현하여야 한다. 우리는 PMR 비율을 {0.5,1.0,3.0,5.0} 중 하나의 값을 취하도록 제한하였다. 따라서 액션은 다음과 같이,

$$a_t = (PMR_{bear}^{bits}(t), PMR_{bull}^{bits}(t), PMR_{GC}^{bits}(t), PMR_{TU}^{bits}(t))$$

의 벡터 형태로 표현된다. 여기에서 $PMR_e^{bits}(t)$는 4비트로서 e번째 지역 투자자의 추천에 배정될 수 있는 4가지 값 중 하나를 가리킨다. 결국, 가능한 액션의 수는 256 이 된다.

이렇게 표현된 상태, 액션의 탐색 공간을 테이블로 표현하면, 테이블 엔트리의 크기는 약 51.2×10^6 개가 된다. Q-학습에서 테이블의 크기가 너무 크면 함수 근사화 모델을 사용하여 Q-테이블을 모델링 하는 것이 일반적인 접근법이다[15].

그림 7은 메타 정책의 Q-학습 알고리즘을 설명한다. 충분한 수의 에피소드를 생성하기 위해, 하나의 에피소드는 거래 개시일과 거래 종료일을 전체 학습 기간 중에서 균등 추출하여 구성하였다. 에피소드 내에서, 에이전트는 Action 함수로부터 액션을 선택하고, 선택된 액션에 대응하는 PMR에 기반하여 주식을 거래한다. 환경으로부터 받은 보상으로부터 Q-테이블의 값을 변경한다. Trade($asset_t$,a_t) 함수는 거래일 t의 시작 자산 $asset_t$와 액션 a_t에 기반한 하루 거래를 시뮬레이션 한다. Trade 함수의 출력은 다음날의 시작 자산으로 사용된다.

보상 함수로서, 우리는 거래 기간의 초기 자산 대비

```
Q(s, a) 를 0으로 초기화;
지정된 에피소드 수만큼 반복
  T1 과 Tn 을 샘플링;
  환경은 asset_T1 과 s_Tn 을 초기화 한다;
  for t = T1 to T_{n-1}
    a_t := Action(s_t)
    asset_{t+1} := Trade(asset_t, a_t)
    if t < T_{n-1} then
      r_t :=0
    else
      r_t := 수익률
    end if
    환경은 s_{t+1} 을 생성한다;
    δ_t := r_t + γ max_{a'} Q(s_{t+1}, a') − Q(s_t, a_t)
    Q(s_t, a_t) := Q(s_t, a_t) + α*δ_t
  end for
```

그림 7 메타 정책을 위한 Q-학습 알고리즘. Action 함수는 Q-정책π에 따라 상태 s_t에 대한 액션 a_t를 리턴한다. Trade 함수는 하루 거래를 시뮬레이션 한다.

이익의 비율인 수익률을 선택하였다.

$$\text{수익률} = 100\times\frac{asset_{T_n} - asset_{T_1}}{asset_{T_1}} \quad (3)$$

에피소드 내의 거래일에는 각 액션의 보상값은 0이며, 에피소드의 종료시에만 수익률이 보상으로 주어진다.

5. 실 험

이 장에서는, Q-학습에 의해 최적화된 메타 정책 을 지닌 투자 시스템 MPT(Trader with Meta Policy)의 성능 비교 실험을 기술한다. 비교를 위해 고전적인 고정 정책을 지닌 두가지 투자 시스템을 포함했는데, 표 6에 각 투자 시스템의 특징을 요약하였다. 각 투자 시스템은 동일한 지역 투자자를 활용하며, 자산의 운용이 각기 다르다.

표 6 투자 시스템의 정책들

이름	의미
trader1	고정 정책, 초기에 분할된 자산 운용
trader2	고정 정책, 통합된 자산 운용
MPT	동적 정책, 통합된 자산 운용

5.1 각 시스템의 정책 및 성능

그림 8과 그림 9는 각각 trader1과 trader2의 자산 운용 방식을 나타낸다. trader1은, 초기 자산을 지역 투자자들 수만큼 균등하게 분할한 후, 분할된 자산이 배타적으로 해당 지역 투자자만을 위해 운용된다. trader1의 자산 할당 정책은 4개의 파라미터,

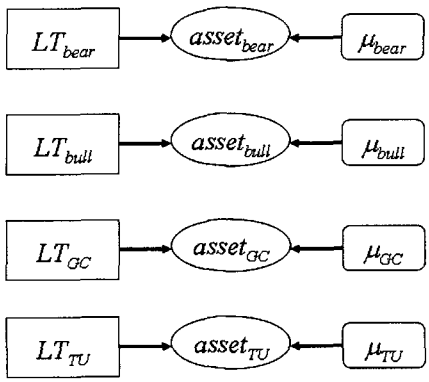

그림 8 trader1의 자산 운용 정책

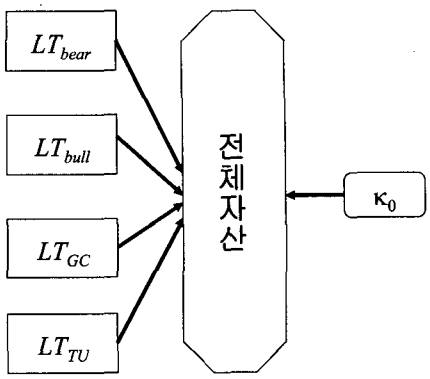

그림 9 trader2의 자산 운용 정책

$\mu_{bear},\mu_{bull},\mu_{GC},\mu_{TU}$

로 구성되었다. 이들은 각 지역 투자자들의 추천에 대한 PMR을 의미한다. 파라미터들은 지역 투자자의 학습 기간과 겹치지 않는 2000년 4월부터 2001년 12월까지의 기간에 대한 시뮬레이션에 대해 최대 수익을 낼 수 있도록 계산되었다. 시뮬레이션은 그림 5의 메타 정책에 따른 거래 과정과 동일하지만, 차이점은 MP 함수를 대신하여 PMR 배정에 trader1의 파라미터 $\mu_{bear},\mu_{bull},\mu_{GC},\mu_{TU}$을 사용한다는데 있다.

trader2는, 초기 자산을 분할하지 않고 통합된 형태로 운용한다. 지역 투자자로부터 추천이 발생할 때마다, trader2는 총자산의 κ_0 비율만큼의 매수를 시도한다. κ_0은 trader1의 파라미터 계산 구간과 동일한 기간에 대해서 시뮬레이션을 통해 최적화 된다. 차이점은 MP 함수 대신 모든 지역 투자자에게 동일한 κ_0의 PMR을 배정하게 된다.

5.2 MPT 학습

MPT는 4장에서 기술한 Q-학습에 의해 구성되었다. 학습 기간은 2000년 4월부터 2001년 12월까지이며, 이는 trader1과 trader2의 파라미터 최적화 구간과 동일하다. MPT 학습을 위해, Q-학습의 세부 파라미터는 다음과 같은 값을 사용하였다. γ는 0.9로 하였고, α는 0.02로 고정하였다. 주식 시장의 불확실성은 학습 기간에서의 과학습(overfitting)을 야기할 수 밖에 없으므로, MPG 학습을 위해, 학습 데이타의 후반 6개월은 검증 기간으로 확보하여 과학습 정도를 감시하는 방법을 택했다. 이를 위해, 각 에피소드의 경험 후, 학습 기간과 검증 기간에 대하여 그림 5의 투자 과정을 따라 거래 성능을 측정하여 그림 10과 같은 학습 로그를 얻었다. 강화학습에서 경험할 에피소드 수는 매우 커서 모두 표기하기 어렵기 때문에, 가로축에 매 1,000 에피소드 지점만을 표기하였다. 세로축은 이에 대응되는 투자 성능

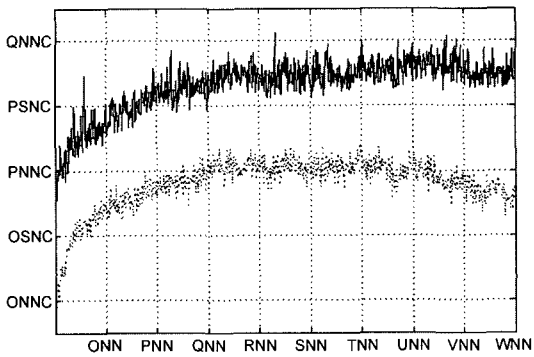

그림 10 Q-학습 경향 분석. 에피소드 경험 후 학습 및 검증 기간에 대한 시뮬레이션된 거래 결과이다. 가로축은 측정 시점인 매 1,000 에피소드 경험을 의미한다. 세로축은 거래 종료시의 수익률이다. 실선은 학습 기간에 대한 수익률, 점선은 검증 기간에 대한 수익률이다.

인 이익률을 나타낸다. 실선은 학습 기간에 대한 이익률이고, 점선은 검증 기간에 대한 이익률이다. 검증 기간에 대한 로그를 살펴보면, 500,000 에피소드 경험 지점에서 약 210% 의 성능에 도달한 후, 점차 성능이 하향 곡선을 나타냄을 알 수 있다. 이는 더 이상의 에피소드 경험을 통한 Q-테이블 갱신은 학습 데이타에 대한 과학습을 이끌게 됨을 의미한다. 따라서, 더 이상의 학습을 중지하고, 그 지점의 Q-테이블을 MPT로 삼았다.

5.3 거래 성능

그림 11은 2002년 1월부터 2003년 5월까지의 테스트 기간에 대하여 세 거래 시스템을 비교한 것이다. 가로축은 거래일을 의미하고, 세로축은 전체 자산을 나타낸다. 각 거래 시스템은 2,500만원의 초기 자산으로 거래를 시작하였다. 점선은 KOSPI 값을 나타내며, 거래 시스템과의 비교의 베이스 역할을 한다. 2002년 1월 2일의 첫

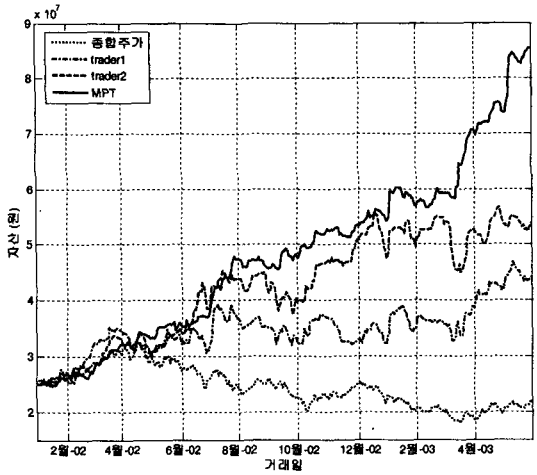

그림 11 세 투자 시스템의 성능 비교. 2002년 1월부터 2003년 5월까지의 거래 로그. 초기 자산 2,500만원으로 거래를 시작했을 때의 자산의 흐름을 나타내었다. 점선(종합주가지수), 점대쉬(trader1), 대쉬(trader2), 실선(MPT)

거래일에 2,500만원처럼 보이도록 KOSPI 값을 스케일링 하였다. 실선은 MPT의 자산 로그, 대쉬 점선은 trader1, 대쉬선은 trader2의 로그이다.

표 7은 각 거래 시스템이 산출한 성능을 요약한 것이다. 수익률은 식 3을 이용하였으며, 상대적 수익률은 다음과 같이 정의된다.

표 7 각 거래 시스템의 산출된 이익 비교

거래 시스템	수익률(%)	상대적 수익률(%)
trader1	76.92	102.49
trader2	115.74	146.92
MPT	241.42	290.46

$$\text{상대적 수익률} = 100 \times \frac{asset_{t_T} - casset_{t_T}}{casset_{t_T}}$$

이 때, $casset_{t_T}$는 최종 거래일의 종합주가 값이다. 즉, 상대적 수익률은 종합주가 대비 수익률을 의미한다.

MPT는 다른 두 거래 시스템의 성능을 확연하게 뛰어 넘고 있다. 17개월의 거래 후에, 자산은 약 6,041만원이 증가하여 약 8,541만원이 되었다. 이는 241.42%의 수익률에 해당한다. trader1은 거래 시스템 중 가장 저조한 성능을 보였다. trader1의 경우, 예측 모델이 많은 수의 추천을 제공했으나, 해당 지역 투자자에게 분할된 자산에 여유가 없어서 그 추천을 모두 매수할 수 없는 반면, 다른 지역 투자자의 자산에는 어느 정도 여유가 있는 경우가 있었다.

trader2의 경우도 제한된 성능을 보였다. 이 경우는, 예측 모델의 출처에 상관없이 추천된 주식의 매수를 시도하기 때문에 trader1의 악조건을 피할 수 있다. 비록 대부분의 기간에서 trader1의 성능을 상위했지만, trader2는 예측 모델간의 상관 관계나, 주식 자금 비율을 고려하지 않는다. 예를 들어, 2002년 9월 무렵, trader1과 trader2는 자산의 많은 감소를 겪지만, MPT는 상대적으로 이를 잘 견뎌냈다. 상황에 따른 적응적인 자산 할당을 통해 위험한 손실을 어느 정도 회피할 수 있고, 또 전체 자산을 크게 상승 시킬 수 있다고 볼 수 있다.

6. 토 론

MPT의 특징을 보다 자세히 살펴보기 위해서, 주식 자금과 추천 정보까지 고려한 종합적인 분석을 그림 12에 보인다. 그림 15의 윗부분은 자산 로그에 대한 그림이다. 중간 부분은 전체 자산 대비 주식 자금의 비율을 나타낸다. 아랫부분은 모든 추천 주식이 매수 된다는 가정하에서 이들의 거래 완료 후의 이익률의 합을 나타낸다. 2002년 4월 이후, 거래소는 하락장 내에 있었다. 2002년 6월말부터 2002년 7월 중순까지, 거래소에는 가파른 상승을 경험한 종목들이 많이 등장했다. 지역 투자자 LT_{bear}는 2002년 6월 27일에 많은 수의 추천을 제공한다. 그 날의 자산 할당 규칙은(5.0, 0.5, 0.5, 1.0) 이었는데, 5.0은 LT_{bear}의 추천 종목에 대한 PMR에 해당한다. 6월 27일에 주식 자금 비율은 약 5% 정도였으므로, 추천된 종목 대부분을 매수할 수 있었다. 이 거래의 결과로, 6월 28일의 주식 자금 비율은 90% 이상으로 상승하였다.

2002년 12월, 거래소는 계속된 하락장으로 재차 국면 전환을 맞는다. 12월 중순 이후 추천 주식 수가 많아졌고 이는 곧 거래 손실이 커질 수 있음을 의미한다. 12월 20일에, LT_{bear}와 LT_{bull}의 추천수가 동시에 많아졌다. 그러나, 이들의 지정된 PMR은 모두 0.5 였다. 즉, 패턴 D_{bear}와 패턴 D_{bull}의 추천 수가 동시에 증가 하고 주식 자금의 비중이 클 때는 매수 대금을 작게 하는게 낫다고 판단한 것이다. 그림 7의 trader1과 trader2를 보면 이 짧은 기간에 자산의 변화가 심함을 볼 수 있고, 손실 또한 컸다. MPT의 경우에는 작은 PMR의 할당으로 인해 주식 자금의 비율이 그리 커지지 않았으며, 상대적으로 작은 손실로 위기를 넘길 수 있었다. 즉, 어느 정도의 손실은 불가피하더라도, 다른 시스템에 비해서는 안정적인 거래를 하고 있음을 알 수 있다. 이런 상황은 trader2가 자산에 심각한 손실을 경험하는 2003년 3월에 더욱 분명히 알 수 있다. MPT는 trader2가 겪는 큰 손실을 최소화 할 수 있었다.

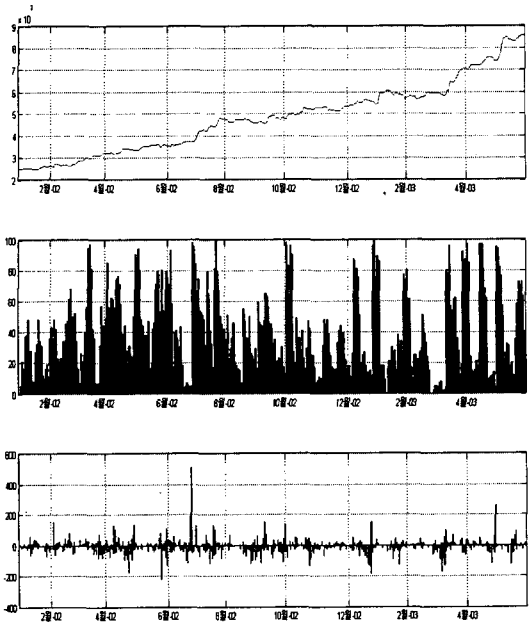

그림 12 MPT의 투자 성향에 대한 상세 분석. 위 그림은 투자 기간 중 자산의 추이를 나타낸다. 가운데 그림은 투자 기간 중 총 자산 대비 주식 자금의 비율(%)을 나타낸다. 아래 그림은 4개의 엔진의 추천들이 전부 거래되었을 경우의 이익의 합(%)을 나타낸다.

물론, MPT가 항상 승리할 수 있는 것은 아니다. 2003년 1월에는 잘못된 추천에 큰 PMR로 매수하여 주식 자금 비율이 증가하여 결과적으로 손실을 입었다. 예측 모델 자체가 완벽할 수 없기 때문에 항상 승리하는 투자를 이끌 수는 없다. 그러나 강화학습을 통해 지역 투자자와 자산 운용을 보다 상위단계에서 고려하는 효율적인 자산 할당 정책을 구성할 수 있었다.

7. 결 론

본 논문에서는, 감독자 학습 기법으로 구성된 지역 투자자를 강화학습 기법 내에서 결합하는 동적 자산 할당 기법을 고안하였다. 예측 모델의 추천 정보와 전체 자산 대비 주식 자금의 정보를 동시에 고려한 강화학습의 상태 표현을 통해 투자금을 상황에 따라 적응적으로 할당하는 메타 정책을 정의 및 최적화하였다.

메타 정책을 이용한 거래 시스템은 테스트 기간 동안의 대한민국 주식 시장에 대한 시뮬레이션에서, 다른 고정 자산 할당 방법에 기반한 거래 시스템들에 비해 큰 이익을 산출했다.

강화학습을 통한 지역 투자자의 결합은, 시간에 따른 이들의 관계 변화로부터 효과적인 의사 결정을 내릴 수 있는 전략을 찾아줄 뿐 아니라, 주식 거래와 같은 복잡한 문제를 해결하는데 감독자 학습으로 구성된 예측 모델의 시너지 효과를 이룰 수 있었다.

참 고 문 헌

[1] S. M. Kendall and K. Ord, Time Series, Oxford, New York, 1997.

[2] E. F. Fama, "Multiperiod Consumption Investment Decisions," American Economic Review, 60, pp. 163-174, 1970.

[3] E. F. Fama and K. R. French, "Dividend Yields and Expected Stock Returns," Journal of Financial Economics, 22, pp. 3-26, 1988.

[4] B. G. Malkiel, A Random Walk Down Wall Street, Norton, New York, 1996.

[5] M. A. H. Dempster, T. W. Payne, Y. Romahi, and G. W. P. Thompson, "Computational Learning Techniques for Intraday FX Trading Using Popular Technical Indicators," IEEE Transactions on Neural Networks, 12(4), pp. 744-754, 2001.

[6] A. Fan and M. Palaniswami, "Stock Selection Using Support Vector Machines," In Proceedings of International Joint Conference on Neural Networks, pp. 1793-1798, 2001.

[7] S. D. Kim, J. W. Lee, J. Lee, and J.-S. Chae, "A Two-Phase Stock Trading System Using Distributional Differences," Proceedings of International Conference on Database and Expert Systems Applications, pp. 143-152, 2002.

[8] E. W. Saad, D. V. Prokhorov, D. C. Wunsch II, "Comparative Study of Stock Trend Prediction Using Time Delay, Recurrent and Probabilistic Neural Networks," IEEE Transactions on Neural Networks}, 9(6), pp. 1456-1470, 1998.

[9] J. W. Lee and J. O, "A Multi-agent Q-learning Framework for Optimizing Stock Trading Systems," Proceedings of International Conference on Database and Expert Systems Applications, pp. 153-162, 2002.

[10] J. Moody and M. Saffell, "Learning to Trade via Direct Reinforcement," IEEE Transactions on Neural Networks, 12(4), pp. 875-889, 2001.

[11] R. Neuneier, "Risk Sensitive Reinforcement Learning," Advances in Neural Information Processing Systems}, pp. 1031-1037, MIT Press, Cambridge, 1999.

[12] J. O, J. W. Lee, and B.-T. Zhang, "Stock Trading System Using Reinforcement Learning with Cooperative Agents," In Proceedings of International Conference on Machine Learning, pp. 451-458, Morgan Kaufmann, 2002.

[13] H. Li, C. H. Dagli and D. Enke, A Comparison Study of Reinforcement Schemes on a Series-

based Stock Price Forecasting Task, IEEE transactions on Neural Networks, Submitted, 2005.
[14] R. S. Sutton and A. G. Barto, Reinforcement Learning : An Introduction. MIT Press, Cambridge, 1998.
[15] K. Hornik, M. Stinchcombe, and H. White, "Multi-layer Feedforward Networks are Universal Approximators", Neural Networks, 2, pp. 359-366, 1989.

오 장 민

1997년 서울대학교 컴퓨터공학 학사 1999년 서울대학교 컴퓨터공학 석사 1999년~현재 서울대학교 컴퓨터공학 박사과정. 관심분야는 Reinforcement Learning, Probabilistic Graphical Model, Kernel Method, Computational Finance

이 종 우

1990년 서울대학교 컴퓨터공학 학사 1992년 서울대학교 컴퓨터공학 석사 1996년 서울대학교 컴퓨터공학 박사 1996년~1998년 현대전자㈜ 정보시스템 사업본부 과장. 1998년~1999년 현대정보기술㈜ 책임연구원. 1999년~2002년 한림대학교 정보통신공학부 조교수. 2002년~2003년 광운대학교 컴퓨터공학부 조교수. 2003년~2004년 아이넉스소프트㈜ 개발이사. 2004년~현재 숙명여자대학교 정보과학부 멀티미디어과학전공 조교수. 관심분야는 Storage Systems, Computational Finance, Cluster Computing, Parallel and Distributed Operating Systems, and Embedded System Software

장 병 탁

1986년 서울대학교 컴퓨터공학 학사 1988년 서울대학교 컴퓨터공학 석사 1992년 독일 Bonn대학교 컴퓨터공학 박사. 1992년~1995년 독일국립정보기술연구소(GMD) 연구원. 1995년~1997년 건국대학교 컴퓨터공학과 조교수. 1997년~현재 서울대학교 컴퓨터공학부 부교수, 인지과학, 뇌과학, 생물정보학 협동과정 겸임. 관심분야는 Biointelligence, Probabilistic Models of Learning and Evolution, Molecular/DNA Computation