

이 재 원 교수지도

석사학위 청구논문

신경망 기반 주가 예측 시스템의
성능 평가를 위한 시각화 도구

2006

성신여자대학교 대학원

전산학전공

정 승 혜

신경망 기반 주가 예측 시스템의 성능 평가를 위한 시각화 도구

이 재 원 교수지도

이 논문을 석사학위 논문으로 제출함.

2005년 11월

성신여자대학교 대학원

전산학전공

정 승 혜

인 준 서

정 승 혜의 석사학위 논문을 인준함.

심사위원 (인)

심사위원 (인)

심사위원 (인)

성신여자대학교 대학원

논문 개요

주식시장은 현대 자본주의 경제체제의 핵심이다. 따라서 주가를 예측하려는 연구는 경영·경제학, 수학·물리학 등 여러 학문 분야에서 연구되어 왔다. 특히 컴퓨터 과학 분야에서의 연구로는 주로 신경망을 비롯한 자동 학습 기법들이 주가 예측 연구에 활용되어 왔다. 그러나 아직까지 이러한 시스템들이 실제 주식시장에 적용되었을 때, 실험과 동일한 수준의 거래 성능을 유지할 수 있는지에 관하여는 아직 연구가 부족한 것으로 판단된다. 이 문제에 있어서 무엇보다 중요한 것은 이러한 시스템의 예측 성능을 보다 객관적이고 체계적으로 평가할 수 있는 평가 기준이 요구된다는 것이다.

이에 본 논문에서는 기존의 주가 예측 시스템들의 실험 성과와 실제 주식시장에서의 거래 성능 간의 괴리를 감소시키기 위해 실제 주식 투자자 입장에서 새로운 성능 평가를 위한 다각적 분석이 가능하도록 시각화 도구를 구현하여 제시하였다. 이를 위해 기존 연구에서 개발, 활용된 신경망 기반 주가 예측 시스템의 종합 개발 도구인 'NNShell'을 거래 성능 평가 체계 중심으로 소개하였으며, 'NNShell'의 성능 평가 방법인 거래 시뮬레이션 결과 파일을 분석하여 상세 내역과 거래 분포도를 그래프로 나타내었다. 본 논문에서 제시된 시각화 도구는 차후 'NNShell'의 추가 기능으로 활용될 수 있다. 사례 분석을 통해 기존의 성능 평가 척도를 통한 성능 평가와 본 논문에서 제시한 시각화 도구의 다각적 분석을 통한 성능 평가를 비교하였으며, 그 결과 시각화 도구를 통하여 예측에 따른 거래의 상세 수익과 거래 빈도 및 분포의 보다 면밀한 비교가 가능하다는 것을 확인하였다.

목 차

논문 개요

I. 서론	1
II. ‘NNShell’에서의 성능평가	3
1. 데이터 가공	3
2. 예측 엔진 생성	7
3. 시뮬레이션	9
III. 개선된 성능 평가를 위한 시각화 도구	12
1. 시각화 도구의 필요성	12
2. 시각화 도구에 필요한 기능	16
3. 시각화 도구의 구현	18
IV. 사례 분석	26
V. 결론	34

참고문헌

ABSTRACT

그림 목차

[그림 2-1] 주가 정보의 데이터베이스 구조 예시 (종목: 삼성전자)	4
[그림 2-2] 피쳐셋 설정 다이얼로그 예시	5
[그림 2-3] 정규화 다이얼로그 예시	6
[그림 2-4] 학습 다이얼로그 예시	7
[그림 2-5] 학습 에러와 밸리데이션 에러의 로그 그래프 예시	8
[그림 2-6] 시뮬레이션 수행 화면 예시	10
[그림 3-1] 시뮬레이션의 결과 파일 예시	12
[그림 3-2] 시각화 도구의 첫 화면	18
[그림 3-3] '상세 그래프 보기' 버튼을 눌렀을 때의 출력 화면	19
[그림 3-4] 슬라이더를 이용하여 막대 넓이, 길이를 가장 크게 설정한 화면	20
[그림 3-5] 슬라이더를 이용하여 막대 넓이, 길이를 가장 작게 설정한 화면	20
[그림 3-6] '간략 그래프 보기' 버튼을 눌렀을 때의 출력 화면	21

[그림 3-7] ‘거래에 따른 달 구분선’을 선택했을 때의 출력 화면	22
[그림 3-8] ‘날짜별로 거래 정렬(누적 출력)’을 선택했을 때의 출력 화면 .	23
[그림 3-9] ‘날짜별로 거래 정렬(중복 시 최대, 최소 출력)’을 선택했을 때의 출력 화면	24
[그림 3-10] 모든 옵션 체크 버튼을 해제했을 때의 화면	25
[그림 4-1] 엔진 A의 간략 그래프	26
[그림 4-2] 엔진 B의 간략 그래프	26
[그림 4-3] 엔진 A의 상세 그래프	27
[그림 4-4] 엔진 B의 상세 그래프	27
[그림 4-5] 엔진 A의 날짜별로 거래 정렬(누적 출력) 화면	28
[그림 4-6] 엔진 A의 날짜별로 거래 정렬(누적 출력) 축소화면	29
[그림 4-7] 엔진 B의 날짜별로 거래 정렬(누적 출력) 축소화면	30
[그림 4-8] 엔진 A의 날짜별로 거래 정렬(최대, 최소 출력) 축소화면	31
[그림 4-9] 엔진 B의 날짜별로 거래 정렬(최대, 최소 출력) 축소화면	31
[그림 4-10] 엔진 A의 중복 추천을 제외한 결과	32
[그림 4-11] 엔진 B의 중복 추천을 제외한 결과	32

I. 서론

현대 자본주의 경제체제의 핵심은 주식시장이다. 따라서 주가를 예측하려는 연구는 여러 학문 분야에서 이루어지고 있다. 경영·경제학, 수학·물리학 등에서도 연구가 이루어지고 있지만, 특히 컴퓨터 과학 분야에서는 인공 신경망(Artificial Neural Networks)을 비롯한 자동 학습 기법들로 주식시장을 예측하고자 하는 시도가 많은 연구자들에 의하여 이루어져왔다[1]. 그러나 아직까지 이러한 주가 예측 시스템들이 실제 주식시장에 적용되었을 때, 실험과 동일한 수준의 거래 성능을 유지할 수 있는지에 관하여는 아직 연구가 부족한 것으로 판단된다. 이 문제에 있어서 무엇보다 중요한 것은 시스템의 예측 성능을 보다 객관적이고 체계적으로 평가할 수 있는 평가 기준이 요구된다는 것이다. 기존의 성능 평가 방법은 시계열 접근법과 거래 시뮬레이션 접근법으로 분류할 수 있는데, 시계열 접근법은 주가의 이동을 시간의 흐름에 따라 분석하는 것으로 고정된 시간 간격에서 예상된 주가와 실제 그 간격 후의 결과를 비교하는 것이며, 거래 시뮬레이션 접근법은 실제 거래와 최대한 유사하게 구현된 시스템을 통해 거래 규칙을 기초로 하여 가상 거래를 통해 결과를 예측하는 방법이다[2]. 이 중 거래 시뮬레이션 접근법이 보다 현실적인 요소들을 잘 반영하고 있는 것으로 판단되어 대부분의 예측 시스템의 경우 거래 시뮬레이션에 의한 성능 평가 척도들을 채택하고 있으나, 실제 주식 투자자들에게 만족스러운 결과를 주고 있는지는 여전히 미지수이다.

이에 본 논문에서는 기존의 주가 예측 시스템 연구들에서 제시된 실험에서의 성능과 실제 주식 시장에서의 거래 성능 간의 괴리를 감소시키기 위해 실

제 주식 투자자 입장에서의 성능 평가가 가능하도록 새로운 시각화 도구를 구현하여 제시하고자 하였다. 이를 위해 기존의 연구에서 개발, 활용된 ‘NNShell’이라는 신경망 기반 주가 예측 시스템의 종합 개발 도구를 거래 성능 평가 체계를 중심으로 소개하였고[3-6], ‘NNShell’의 성능 평가 방법인 거래 시뮬레이션 결과 파일을 분석하여 상세 내역과 거래 분포도를 그래프로 나타내었다. 본 논문에서 제시된 시각화 도구는 거래 시뮬레이션의 결과를 다각적으로 분석하기 위한 기능들로 구성하였고, 차후 ‘NNShell’의 추가 기능으로 활용될 수 있다. 실제 사례 분석을 통해 기존 성능 평가 척도를 통한 결과 성능과 본 논문에서 제시하는 시각화 도구를 이용한 결과 성능을 비교하고 이에 따른 분석을 실시하였다.

이 논문의 구성은 다음과 같다. 제 2장에서는 기존 ‘NNShell’의 주요 기능인 데이터 가공, 주가 추세 예측 엔진 생성, 시뮬레이션에 이르는 과정을 소개하고, 제 3장에서는 본 논문에서 제시한 시각화 도구의 필요성, 이에 따라 요구되는 기능에 대하여 설명한다. 그리고 실제 구현한 시각화 도구를 제시한다. 제 4장에서는 본 논문에서 제시한 시각화 도구를 이용하여 실제 주가 추세 예측 엔진들의 성능을 비교·평가하고, 마지막으로 제 5장에서 결론과 추후 과제를 제시한다.

II. ‘NNShell’에서의 성능 평가

이 장에서는 기존의 연구에서 개발, 활용된 주식 거래 시스템의 개발을 위한 종합 개발 도구인 ‘NNShell’에 관하여 소개한다[3-6].

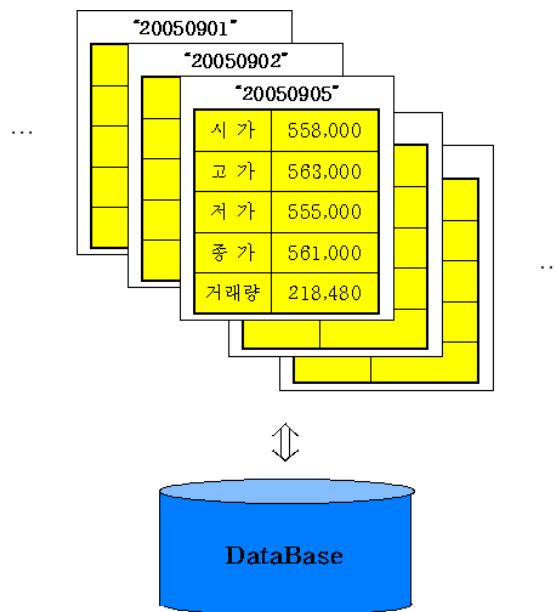
‘NNShell’은 신경망을 기반으로 한 주가 예측 시스템으로, 데이터베이스에 저장된 원 데이터(raw data)를 가공하여 기계학습을 수행하며, 파라미터들의 조합으로 하위(sub) 시스템인 최적 상태의 주가 추세 예측 엔진을 생성한다. 엔진의 성능 평가를 위해 가상 거래(simulation)를 수행하여 앞서 생성된 엔진의 성능을 측정하는 통합된 개발 도구이다. 이 단계들을 세단계로 나누어 본 장에서 자세히 설명하겠다.

1. 데이터 가공

한국 주식시장 각 종목의 주가 정보를 저장하여 주가 예측 시스템의 입력으로 사용할 서버로 구축한다. 주가 정보를 매일의 시가, 종가, 고가, 저가, 거래량의 5가지 인자로 표현한 원 데이터(raw data)를 데이터베이스에 저장하여 사용한다[7]. [그림 2-1]은 이러한 주가 정보의 데이터베이스 구조 예시를 보여준다. 이 주가 정보는 다음과 같다.

1) 시가 : 하루 중 거래가 시작되고 나서 최초로 성립된 가격.

- 2) 종가 : 하루 동안 열린 장에서 가장 마지막으로 거래가 성립된 가격.
- 3) 고가 : 하루 중 가장 높은 값으로 거래가 성립된 가격.
- 4) 저가 : 하루 중 가장 낮은 값으로 거래가 성립된 가격.
- 5) 거래량 : 하루 동안 매매가 이루어진 주식의 수량.

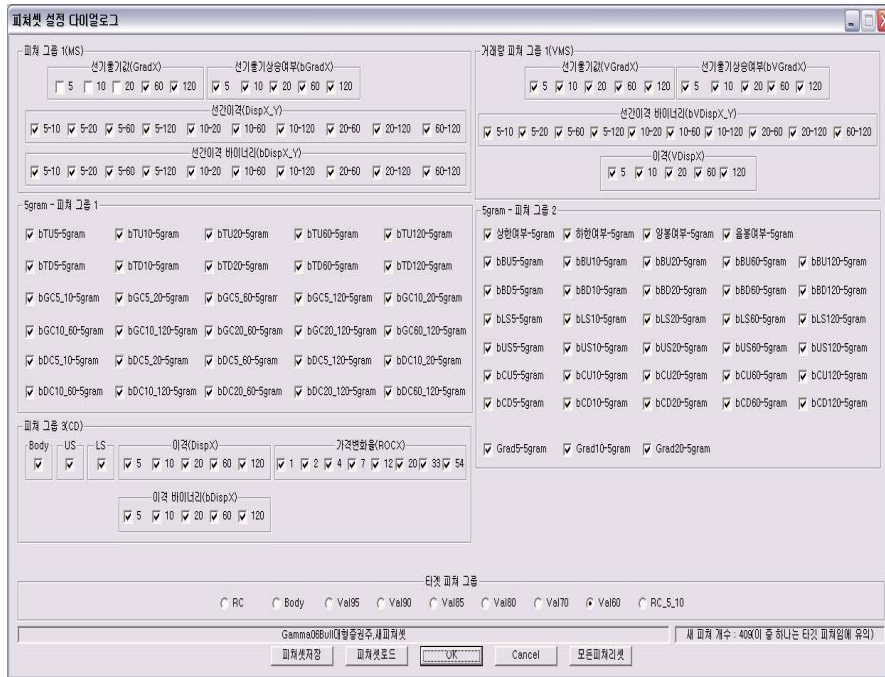


[그림 2-1] 주가 정보의 데이터베이스
구조 예시 (종목: 삼성전자)

이렇게 데이터베이스에 저장된 데이터를 'NNShell'의 입력 값 형태(피쳐 셋)로 가공한다.

먼저, 추출할 데이터의 구성을 설정한다. 이는 다음 예측 엔진 생성 단계

에서 이루어질 학습 과정의 자료로 사용할 데이터의 요소를 말한다. [그림 2-2]는 피쳐셋 설정 다이얼로그를 예시한 것으로, 수많은 체크박스들은 가격 변화율, 주식 차트의 막대 색깔 등 원 데이터로 표현 가능하여 학습에 사용할 요소들을 설정 가능하게 한다.



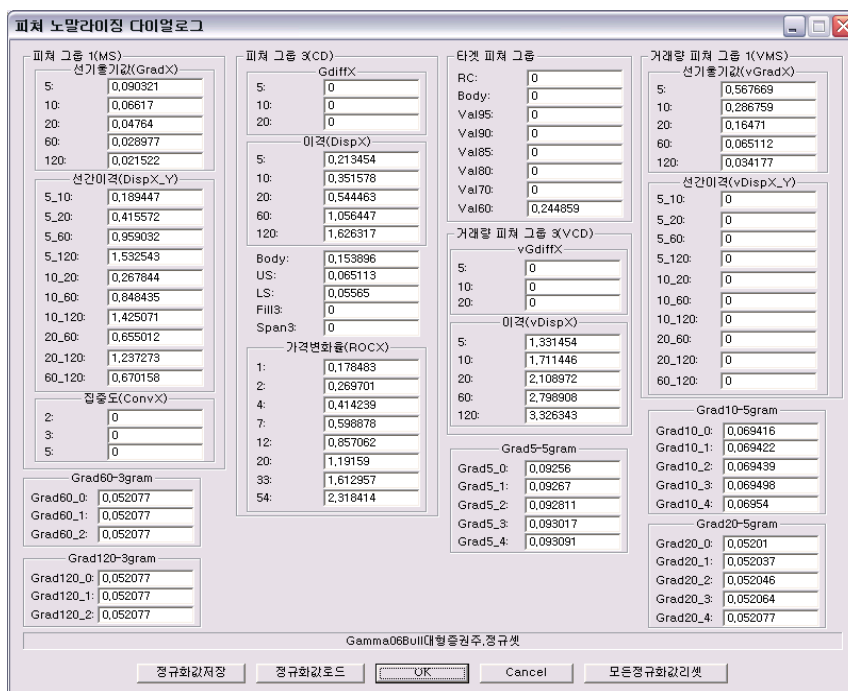
[그림 2-2] 피쳐셋 설정 다이얼로그 예시

추출 데이터 구성을 설정한 후에, 이 데이터를 사용하여 신경망을 통한 학습을 수행할 기간과 밸리데이션을 수행할 기간을 설정한다. 밸리데이션은 학습의 결과 성능을 검증하는 과정으로서, 학습 수행 기간 동안 학습한 것을 밸리데이션 기간에 테스트하여 최적의 학습 성능을 나타내는 시점을 찾은 후에 이 시점까지 재학습을 수행할 것이다.

이어서 한국 주식시장의 모든 종목에 관한 데이터를 학습 자료로 사용할

것인지, 아니면 코스닥 증권시장의 종목들만 학습 자료로 사용할 것인지 등을 종목 후보군을 통해 설정하고, 특정 분야의 종목들(예를 들면 건설주, 통신주 등)만 학습 자료로 사용할 것인지를 필터링 항목을 통해 설정한다.

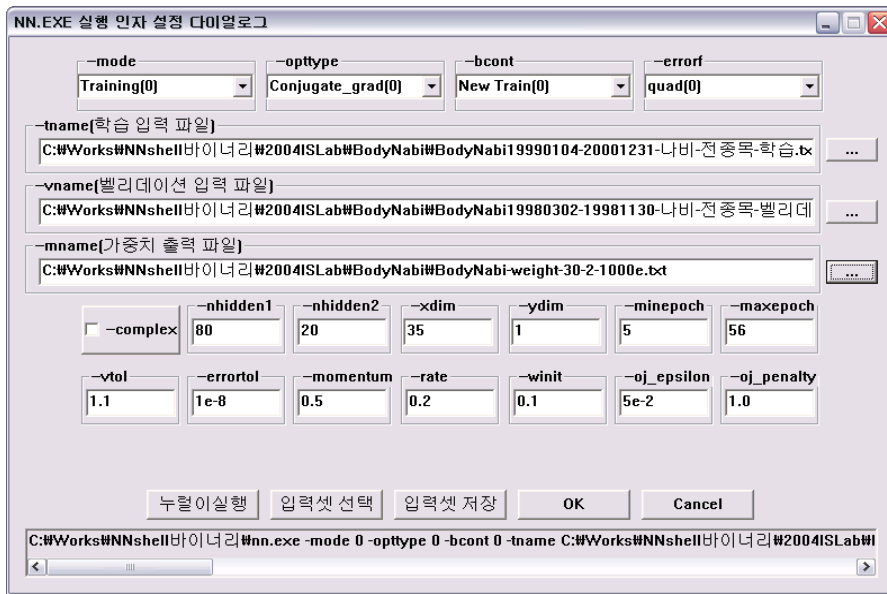
마지막으로 이제까지 설정한 형식에 따라 추출할 데이터를 신경망 입력에 적합한 -1에서 1까지의 범위에 맞도록 배정해주는 기준을 정규화(normalization)를 통해 설정하면, 데이터 가공이 완료되고 피쳐셋이 완성된다. 이러한 정규화의 실행 화면은 [그림 2-3]과 같다.



[그림 2-3] 정규화 다이얼로그 예시

2. 예측 엔진 생성

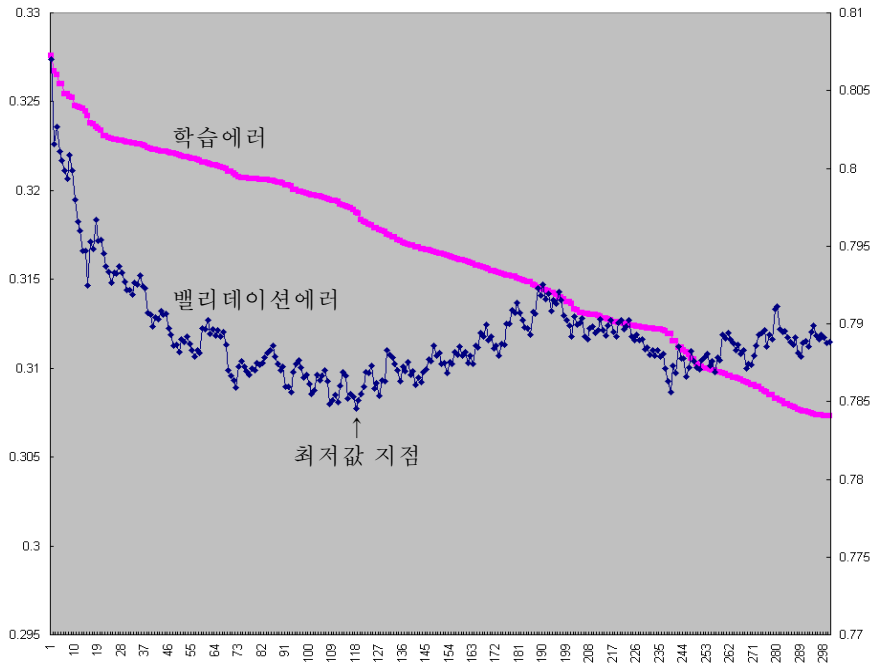
앞서 설정한 피쳐셋으로 학습과 밸리데이션에 필요한 입력 파일을 생성하고, 이를 입력으로 하여 신경망을 통한 학습을 진행한다. 이후 연속으로 밸리데이션을 수행한다. [그림 2-4]는 학습 다이얼로그를 예시한 것으로, 학습 입력 파일과 밸리데이션 입력 파일, 학습 후 생성할 가중치 출력 파일을 지정하고 학습의 기반인 신경망과 학습의 타입을 설정한다.



[그림 2-4] 학습 다이얼로그 예시

학습과 밸리데이션을 진행하면서 각각의 에러 로그 그래프([그림 2-5] 참조)를 확인한다. 그래프의 가로축은 학습을 반복한 횟수(epoch)이고, 좌측의 세로축은 학습 에러의 수치이며 우측의 세로축은 밸리데이션 에러의 수치이다. 학습을 여러 번 반복할수록 학습 에러는 계속 낮아지지만, 밸리데이션

에러는 일정 횟수의 반복 학습 때까지만 줄어들고, 그 이후에는 오히려 상승하게 된다.



[그림 2-5] 학습 에러와 밸리테이션 에러의 로그 그래프 예시

[그림 2-5]의 경우를 보면, 밸리테이션 에러가 학습을 120회 반복했을 때까지는 줄어들지만, 120회 이상으로 학습이 진행되면 다시 상승하게 된다. 따라서 밸리테이션 에러가 가장 낮을 때의 지점이 가장 학습 효과가 좋은 것으로 판단하여 그 지점의 학습 반복 횟수를 찾아 그 횟수만큼 재학습을 수행한다. 재학습을 통해서 에러가 가장 낮은 시점의 학습 효과를 보유했 수 있게 된다. 이러한 학습 과정의 결과물로 가중치의 변화 기록 파일이 생성되는데, 최적 지점까지의 재학습을 하는 과정에 생성된 가중치 파일을 기반으로 주가 추세 예측 엔진이 생성된다.

3. 시뮬레이션

앞서 설명한 과정으로 주가 추세 예측 엔진이 생성되었는데, 이렇게 생성된 예측 엔진의 성능을 평가하는 과정이 필요하다. 주가 추세 예측 엔진의 성능을 평가하기 위한 방법으로는 시계열 접근법과 거래 시뮬레이션 접근법으로 분류할 수 있는데, ‘NNShell’에서는 시뮬레이션 접근법을 수행하였다 [8]. 앞의 II-2절에서 예측 엔진 생성의 결과물로 해당 예측 엔진의 예측치를 출력하는 예측치 파일이 생성되는데, 시뮬레이션은 이 예측치 파일을 기초로 하여 실제 거래와 최대한 유사하게 구현된 시스템을 통해 가상 거래를 수행하는 과정이다. 본 과정의 실행 화면은 [그림 2-6]과 같다.

예측치 파일에는 예측 엔진이 생성한 예측들이 출력되어 있다. 예측 엔진은 앞서 수행하였던 학습에 따라 수익을 가져올 것이라고 예상되는 주식 종목, 거래 수행 날짜, 예측치를 출력하게 되는데, 이 예측치는 예측 엔진이 출력한 예측의 예상 정확도를 나타내는 수치이다. 이에 따라 시뮬레이션 수행 시에 특정 예측치 이상일 때에만 그 예측을 고려한다는 예측치의 범위를 설정할 수 있다. 시뮬레이션의 예측치 범위를 높게 설정할수록 수익을 내는 정확도는 높아지지만 고려할 수 있는 예측의 수는 줄어들 것이며, 예측치 범위를 낮게 설정할수록 예측의 정확도는 낮아지지만 고려할 수 있는 예측의 수는 늘어날 것이다. 시뮬레이션이 수행될 때, 예측 엔진은 이러한 예측치 범위 설정과 매수할인율, 이익실현율, 손절율 등의 예측 엔진의 중요 요소가 되는 파라미터들을 조합하여 어떤 종목을 어느 날짜에 어느 가격을 주고 매수하라는 ‘추천’을 생성하게 되는데, 가상거래는 이 ‘추천’에 따라서 이루어지게 되고, 여기서 최고의 가상 거래 수익을 내는 파라미터들의 조합을 탐색한다[9].

성이 크다. 이는 본 예측은 이익을 가져올 가능성이 낮기 때문에 일정 비율 이하로 매수가 이루어질 때만 거래를 성사시키라는 의미이다.

그리고 이익실현율은 매수하여 현재 보유하고 있는 주식의 가격이 상승하고 있다면, 어느 시기에 매도할 것인지를 결정하는 파라미터이다. 예를 들어 이익실현율 구간 설정 값이 '12'라면, 이는 매수한 당시 주가의 12%가 오르면 목표를 달성하였으니 이제 매도하라는 의미이다.

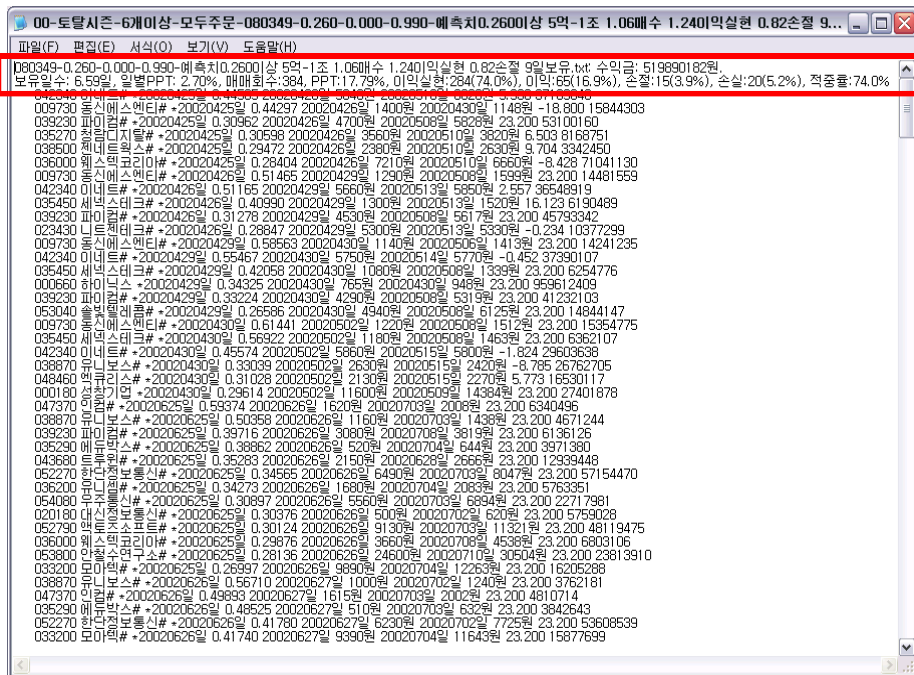
이에 반해 손절율은 현재 보유하고 있는 주식의 가격이 하락하고 있을 때, 어느 시기에 매도할 것인지를 결정하는 파라미터이다. 예를 들어 손절율 구간 설정 값이 '-10'이라면, 이는 매수한 당시 주가의 10%가 하락하면 매도하라는 의미이다. 매수와 매도의 추천에 관한 요소로는 이익실현율, 손절율 외에도 보유일수가 있다.

보유일수는 현재 보유하고 있는 주식의 가격이 이익실현율에 미치지 못하였으며, 그렇다고 손절율까지 하락하지도 않은 가격일 때, 거래일로부터 보유일수를 채우면 매도하라는 의미의 요소이다. 보유일수는 시뮬레이션을 실행할 때 별도로 지정하지는 않고, 엔진 스스로 모든 조합을 통해 최적의 보유일수를 선택한다.

Ⅲ. 개선된 성능 평가를 위한 시각화 도구

1. 시각화 도구의 필요성

‘NNShell’의 시뮬레이션이 수행되고 나면 [그림 3-1]과 같은 시뮬레이션 결과(가상 거래 내역)를 얻게 된다.



[그림 3-1] 시뮬레이션의 상세 내역이 나타나 있는 결과 파일 예시

시뮬레이션 결과 파일에는 상세한 가상 거래 내역과 함께 시뮬레이션 접근법에서의 성능 척도에 따라 측정된 성능이 제시된다. 이 성능 척도들은 다음과 같다[10].

1) PPT (Profit Per Trade)

거래 당 순수 수익을 말한다. 이는 거래 수수료와 거래세를 제외한 순수한 수익이며, 거래 수수료와 거래세는 거래 대금의 약 0.35%가 부과된다. 거래 수수료는 증권사에 납부되는 금액으로 매수, 매도 시 각각 부과되며 거래세는 주식을 매도할 경우에만 부과된다. 'NNShell'에서는 다량의 주식을 매수했을 때 나타날 수 있는 주가 변동의 가능성 등 실제 거래 시 나타날 수 있는 현실과의 문제를 고려하여 투자자가 주가에 가장 큰 영향을 주었을 때를 기준으로 하여 0.8%로 지정하여 적용한다.

$$PPT(\%) = \frac{\text{총 수익}(\%)}{\text{거래 횟수}} - 0.8$$

2) 일별 PPT

날짜 별 수익을 말한다. PPT는 수익을 얻기까지의 소유된 기간에 대한 고려가 없으므로, 이에 대한 시간 효율을 고려한 것이다.

$$\text{일별 } PPT(\%) = \frac{PPT(\%)}{\text{보유 일수}}$$

3) 이익실현

시뮬레이션 실행 시 설정한 거래 정책인 이익실현율을 달성하여 매도한

거래의 수이다. [그림 3-1]의 경우 이익실현율은 1.24로 설정되어 있으므로 24%에서 거래 세 0.8%를 제외한 23.2%의 이익을 달성한 거래가 284건이고 이는 전체 거래의 74.0%이다.

4) 이익

비록 목표치인 이익실현율을 달성하지는 못했지만, 보유일수가 만료되어 매도한 거래가 양수의 이익을 얻은 거래의 수이다.

5) 손절

시뮬레이션 실행 시 설정한 거래 정책인 손절율만큼 주가가 하락하여 매도한 거래의 수이다. [그림 3-1]의 경우 손절율은 0.82로 설정되어 있으므로 18%에서 거래세를 포함한 18.8%의 손실을 입은 거래가 15개로 전체의 3.9%이다.

6) 손실

손절율만큼 하락하지는 않았으나 보유일수가 만료되어 매도한 거래의 수익률이 음수인 거래의 수이다.

7) 적중률

엔진이 예측한 결과를 실제 주식시장에 적용시켰을 때의 정확도이다. 적중률의 수치는 전체 거래 중 이익실현을 달성한 거래의 비율로 측정한다.

8) 상세 거래 내역

가상 거래가 이루어진 종목 번호, 종목명, 추천일, 예측치, 매수일, 매수

가격, 매도일, 매도 가격, 본 거래로 발생한 수익률이 차례대로 나열되어 있다. 이는 사용자로 하여금 자세한 거래내역을 볼 수 있게 하여 예측 엔진의 성향을 파악할 수 있게 한다.

그러나 성능 척도에 따른 성능은 단순 수치로만 나와 있어 여러 엔진의 비교가 힘들다. 한 발 더 나아가서 투자자의 심리상태를 고려한 척도, 즉 기간에 따른 거래 분포도나 편차가 큰 이익과 손실보다는 작지만 지속적인 이익의 존재 유무 등을 나타내기에는 부족하다. 이는 상세 거래 내역을 통해 보완될 수 있으나, 복잡한 텍스트로 인해 인식하기에 힘든 점이 있는 것이 사실이다. 따라서 본 논문에서는 사용자가 엔진의 성능을 보다 쉽게 파악하여 평가할 수 있도록 시뮬레이션 결과 파일을 그래프로 나타내는 시각화 도구를 구현하여 종합 개발 도구인 'NNShell'의 추가 기능으로 활용함으로써 이의 유용성과 편리성을 높이고자 한다.

2. 시각화 도구에 필요한 기능

본 절에서는 시뮬레이션 결과 파일을 그래프로 나타내는 시각화 도구를 구현할 때, 이 도구에 필요한 기능들에 대해 알아보겠다.

첫째로, 성능 평가 요소들의 그래프화가 필요하다. [그림 3-1]의 시뮬레이션 결과 파일에서 두 번째 줄을 보면 PPT, 적중률 등의 기존 성능 평가 요소들의 수치가 나타나 있는 것을 볼 수 있다. 이는 아래 거래 내역들을 다 살펴보지 않더라도 해당 예측 엔진의 성능을 어느 정도 평가할 수 있지만, 단순 수치로만 나와 있어서 해당 성능을 다른 엔진들과 비교하여 평가하려면 주의력을 요한다는 단점이 있다. 따라서 기존 성능 평가 요소들에 대한 결과를 한눈에 보고 평가할 수 있는 그래프가 필요한 것으로 보인다.

둘째로, 시뮬레이션 결과 파일에 나타나 있는 모든 가상 거래 내역의 그래프화가 필요하다. 실제로 [그림 3-1]의 시뮬레이션 결과 파일의 세 번째 줄부터 가상 거래 내역이 나와 있는데, 각 거래의 이익 또는 손실 비율뿐만 아니라 거래 종목, 추천일, 매수대금 등이 한꺼번에 제시되어 한눈에 보기에 불편하다. 그러므로 성능 평가에 필요한 가상 거래의 손익 비율만을 그래프로 나타내어 인식을 용이하게 할 필요성이 있다.

셋째로, 가상 거래 추천일의 그래프 반영이 필요하다. 물론 가상 거래의 손익도 중요하지만, 실제 투자자의 입장에서 본다면 투자 시기의 중요성도 빼놓을 수 없다. 거래 추천이 한 시기에 몰려있으면 투자금의 부족으로 모든 거래를 성사시킬 수 없고, 어느 시기에는 한동안 추천이 나오지 않는다

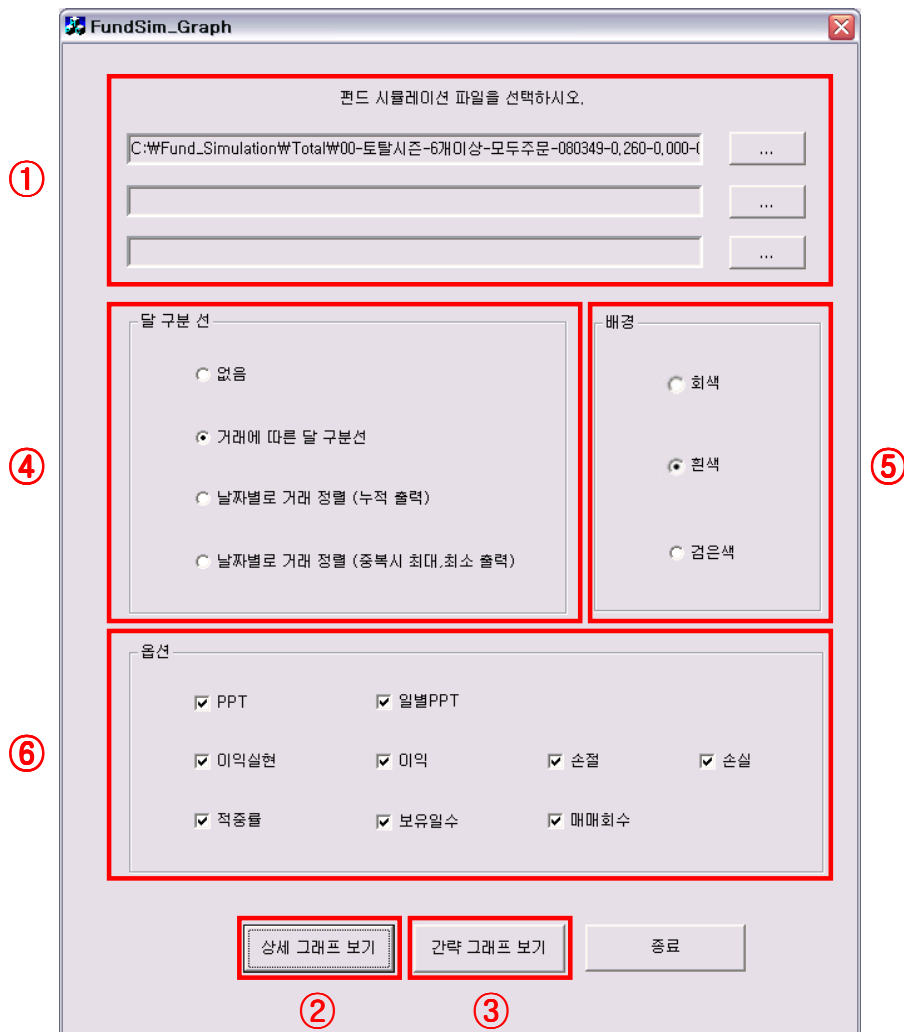
면 투자자의 입장에서 더 이상 엔진을 신뢰하지 않을 가능성이 있기 때문이다. 따라서 손익 비율뿐만 아니라 거래 시기도 중요한 요소이므로 그래프에 반영되어야 할 것으로 보인다.

넷째로, 여러 엔진들의 비교·평가를 위한 기능이 필요하다. 예측 엔진 성능 평가는 한 엔진의 평가를 위한 것이지만, 아울러 여러 엔진을 비교하고 평가하여 성능이 좋은 엔진을 선택하기 위함이기도 하다. 그러므로 두 개 이상의 엔진을 비교하여 평가할 수 있는 기능이 필요할 것으로 판단된다.

위에서 시각화 도구에 필요한 기능들을 살펴보았다. 이와 같은 기능들을 모두 충족시키면서 효율적인 인터페이스를 갖춘, 실제 투자자가 실용적으로 활용할 수 있는 시각화 도구의 구현에 대해 다음 절에서 서술하고자 한다.

3. 시각화 도구의 구현

본 논문에서 구현한 시각화 도구의 첫 화면은 [그림 3-2]와 같다. 이 도구에 포함된 각각의 기능을 살펴보면 다음과 같다.



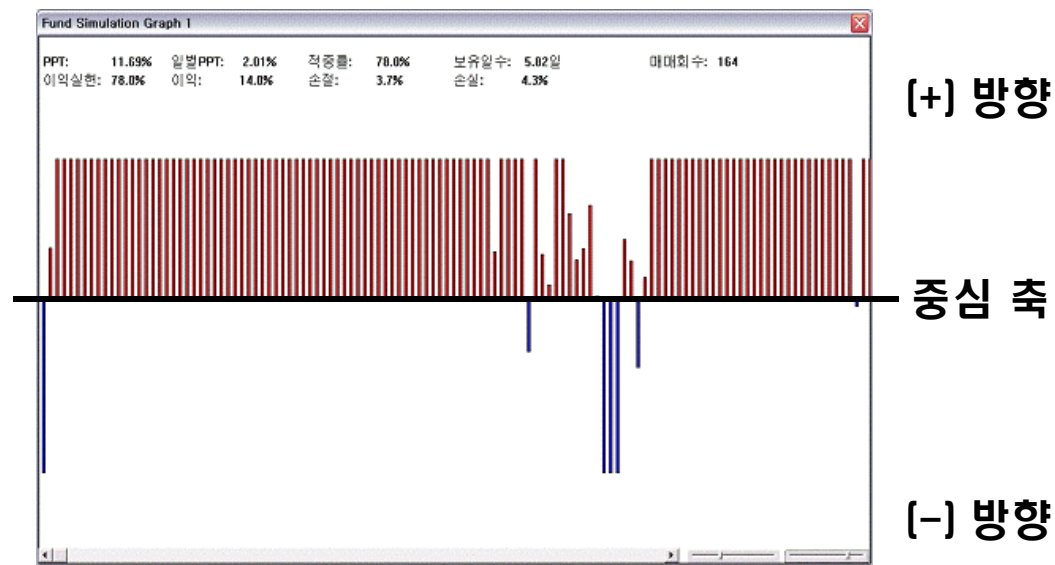
[그림 3-2] 시각화 도구의 첫 화면

① 파일 선택

그래프로 나타낼 시뮬레이션 결과 파일을 선택한다. ‘...’ 버튼을 눌러 해당 텍스트 파일을 선택하면 해당 경로가 나타난다. 여러 엔진을 동시에 비교, 평가할 수 있도록 파일 3개까지 선택 가능하도록 하였다.

② 상세 그래프 보기

선택한 시뮬레이션 결과 파일로 그래프를 그려 시뮬레이션 내역을 세부적으로 볼 수 있게 하였다 [그림 3-3]은 이 출력의 예시 화면이다.

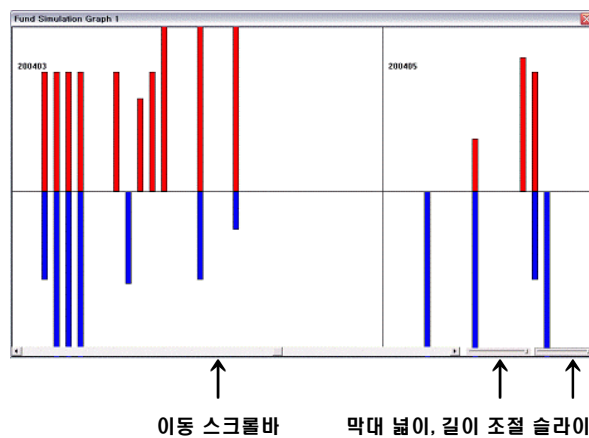


[그림 3-3] ‘상세 그래프 보기’ 버튼을 눌렀을 때의 출력 화면

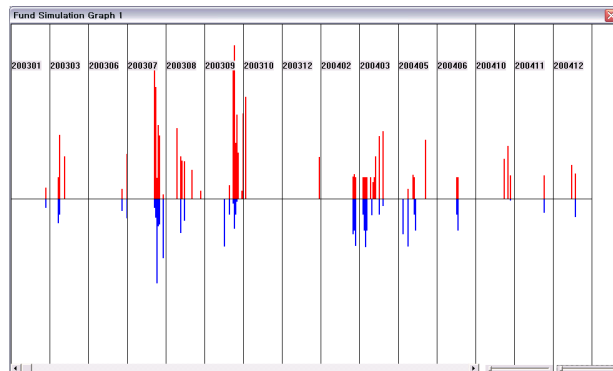
그래프에서 각 가상 거래 내역을 좌측 끝부터 순차적으로 배열하였고, 거래의 가장 중요한 결과인 수익률만큼 막대 길이를 나타내었다. 중앙의 가로축을 기준으로 하여 수익률이 양수인 거래는 그만큼의 길이를 빨간색 막대로 중심 축 상단으로([+] 방향) 나타내었고, 수익률이 음수인 거래는 손실만

큼의 길이를 파란색 막대로 중심 축 하단으로([-] 방향) 나타내었다.

하단 좌측의 스크롤바로 첫 거래부터 마지막 거래까지 볼 수 있게 하였고, 하단 우측의 두 개의 슬라이더를 이용하여 그래프의 막대 넓이와 길이를 각각 조절할 수 있게 하였다. [그림 3-4]는 막대 넓이와 길이를 가장 크게 설정한 화면으로서 자세한 거래의 비교가 가능하도록 하였고, [그림 3-5]는 막대 넓이와 길이를 가장 작게 설정한 화면으로서 전체적인 거래의 분포 비율을 쉽게 볼 수 있도록 하였다.



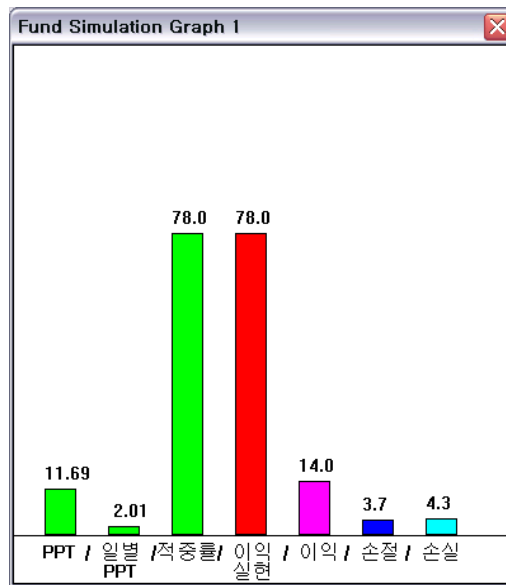
[그림 3-4] 슬라이더를 이용하여 막대 넓이, 길이를 가장 크게 설정한 화면



[그림 3-5] 슬라이더를 이용하여 막대 넓이, 길이를 가장 작게 설정한 화면

③ 간략 그래프 보기

앞 장에서 설명한 기존 성능 척도로 전체적인 성적을 그래프로 나타낸다. 이를 통해 자세한 거래 내역은 볼 수 없지만 ‘상세 그래프 보기’와 함께 사용할 수 있다. [그림 3-5]는 ‘간략 그래프 보기’의 예시이며, 해당 성능 척도의 성적만큼 막대 길이를 나타내었고 각 막대 아래에 어떤 척도를 의미하는지를 나타내었다.



[그림 3-6] ‘간략 그래프 보기’ 버튼을 눌렀을 때의 출력 화면

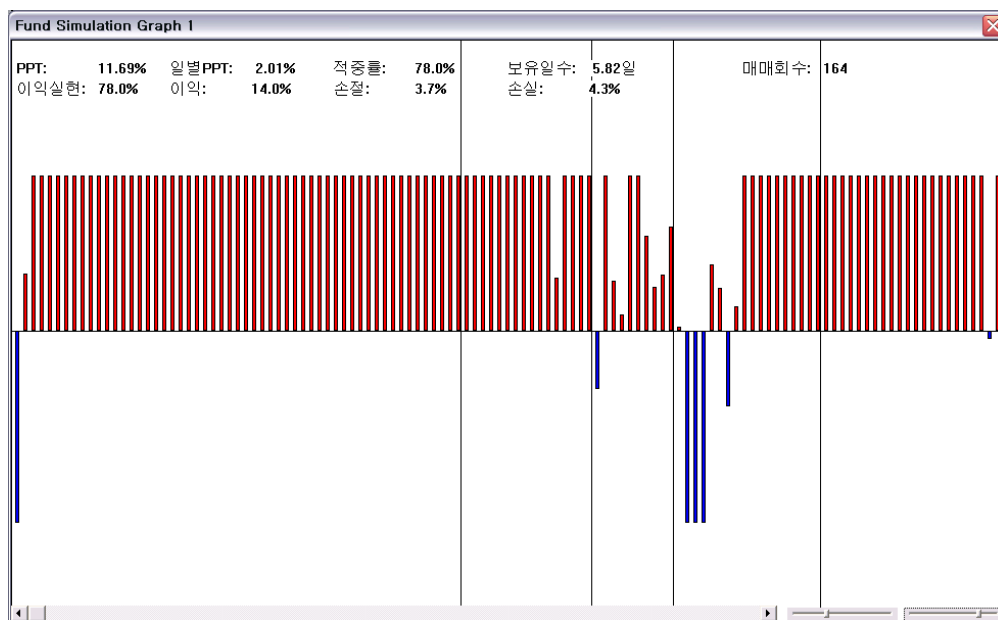
④ 달 구분선

4개의 라디오 버튼을 두어 모드를 선택할 수 있게 하였다.

첫 번째 ‘없음’ 라디오 버튼은 부가적인 기능 없이 시뮬레이션에서의 각각의 거래를 그래프로 나타낸다. 이는 [그림 3-3]의 예시 화면과 같다.

두 번째 ‘거래에 따른 달 구분선’ 라디오 버튼은 첫 번째 라디오 버튼을

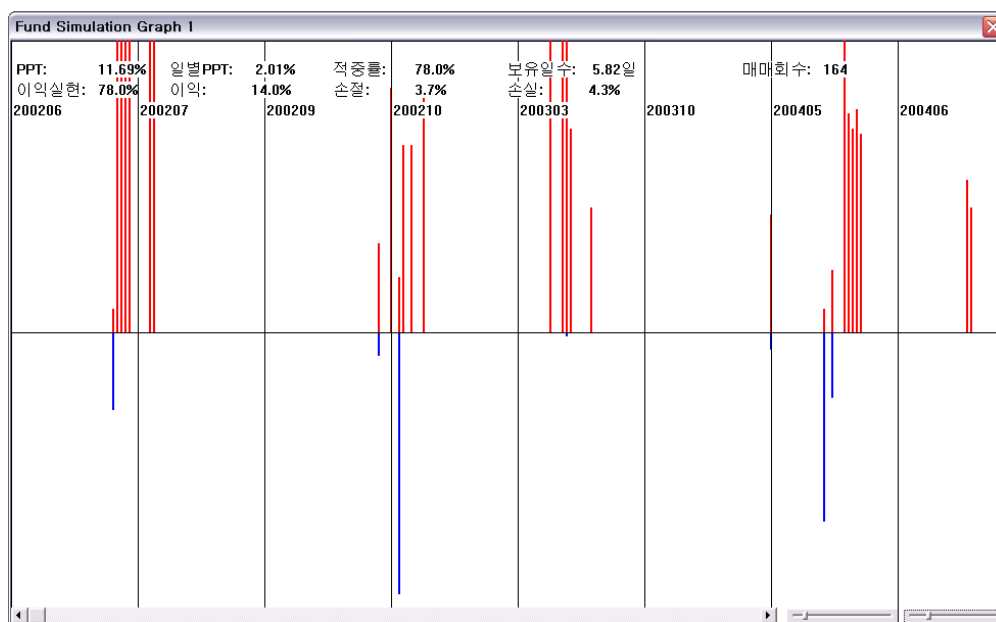
눌렀을 때의 그래프에 거래한 달이 바뀔 때마다 구분선을 나타내어 한 달 동안 이루어진 가상 거래의 횟수를 알 수 있게 한다. [그림 3-7]은 이의 예시 화면이다.



[그림 3-7] '거래에 따른 달 구분선'을 선택했을 때의 출력 화면

세 번째 '날짜별로 거래 정렬(누적 출력)' 라디오 버튼은 거래가 이루어진 각 달의 간격을 같게 유지하고, 그래프 막대의 위치를 날짜에 따라 나타낸 것이며 [그림 3-8]은 이의 예시를 보여주는 그래프이다. 예를 들어 2005년 9월 5일에 이루어진 가상 거래라면, 2005년 9월에서 5일에 해당하는 위치에 그래프 막대가 나타나게 된다. 여기에 하루에 추천이 두 개 이상이라면, 수익은 수익끼리 합하고 손실은 손실끼리 합하여 각각의 절대 값을 그래프 막대의 길이로 나타낸다. 예를 들어 2005년 9월 5일에 추천이 4개가 생성되어 거래를 한 결과 각각 수익이 10%, -5%, -8%, 12%가 나왔다고 가정하면,

해당 날짜의 위치에 빨간색 막대는 수익인 10%와 12%의 합인 22% 만큼의 길이로 그려지고, 같은 위치에 파란색 막대는 -5%와 -8%의 합인 -13% 만큼의 길이만큼 중심축에서 아래로 그려지게 된다. 이렇게 하루에 추천이 단 한 개가 아니라 두 개 이상이 나올 경우 본 논문의 이후 부분에서는 ‘중복 추천’이라고 부르기로 한다.

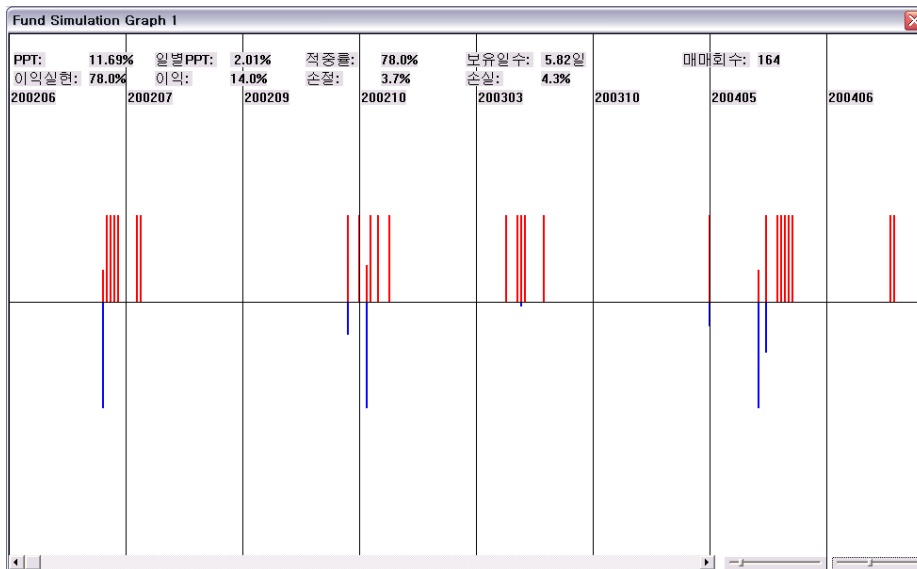


[그림 3-8] ‘날짜별로 거래 정렬(누적 출력)’을 선택했을 때의 출력 화면

네 번째 ‘날짜별로 거래 정렬(중복 시 최대, 최소 출력)’ 라디오 버튼은 해당 거래 날짜 위치에 그래프를 그리는 것은 세 번째 라디오 버튼과 같으나, 중복 추천이 나오게 되면, 그날의 총 거래 중 가장 큰 수익과 가장 큰 손실만을 나타낸다. [그림 3-9]에 중복 추천을 제외하고 최대, 최소만 출력한 예시를 나타내었다. 예를 들어 2005년 9월 5일에 추천이 4개가 생성되어 거래를 한 결과 각각 수익이 10%, -5%, -8%, 12%가 나왔다고 가정하면, 해당

날짜의 위치에 빨간색 막대는 수익인 10%와 12% 중에서 큰 값인 12% 만큼의 길이로 그려지고, 같은 위치에 파란색 막대는 -5%와 -8% 중에서 큰 값인 -8% 만큼의 길이만큼 중심축에서 아래로 그려지게 된다.

여기서 세 번째와 네 번째 라디오 버튼을 구별한 이유는, 예측 엔진을 시뮬레이션하면 중복 추천이 나오는 경우가 많은데 일반 투자자의 입장에서 보면 거래 대금이 무한대가 아니라 한정되어 있으므로 여러 개의 추천이 나온다고 해서 그 모든 추천에 투자하는 것은 현실적으로 불가능하기 때문이다. 따라서 본 시각화 도구 구현의 목적인 투자자들의 심리 상태를 고려하는 성능 평가가 이루어지기 위해서는, 하루에 여러 개의 추천이 나왔을 때 투자 자금이 무한대라고 가정하고 이를 모두 거래한 결과와 투자자금이 한정되어 있어서 추천 중에 선택적으로 거래를 했을 때 나올 수 있는 최대의 수익과 최대의 손실을 함께 보여주는 것이 유용할 것이라고 판단된다.



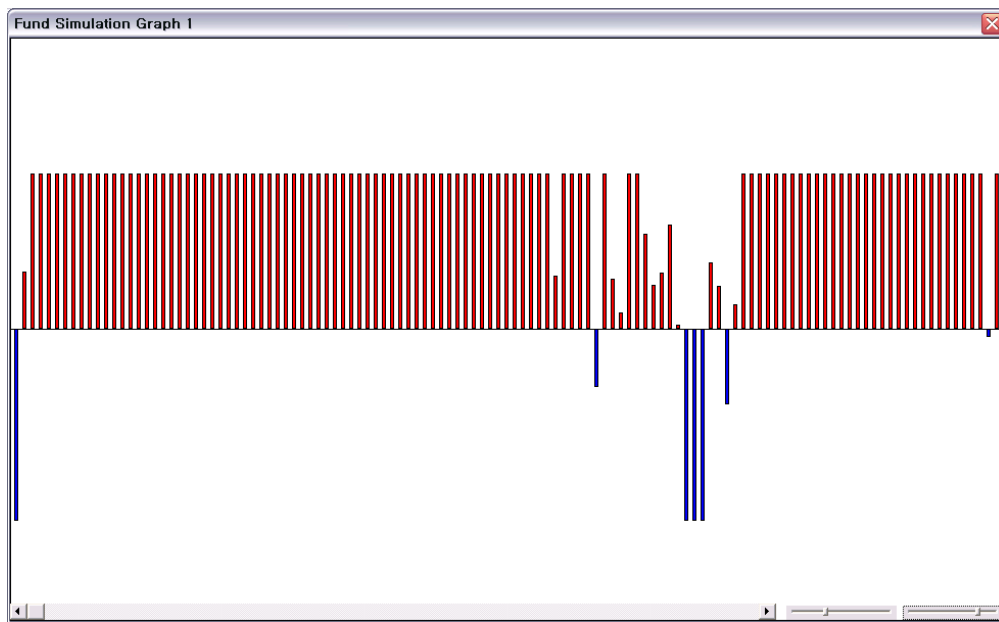
[그림 3-9] '날짜별로 거래 정렬(중복 시 최대, 최소 출력)'을 선택했을 때의 출력 화면

⑤ 배경

사용자의 기호에 따라 배경을 3가지 색으로 선택할 수 있도록 하였다.

⑥ 옵션

9개의 체크 버튼을 사용하여 그래프에 기존 성능 평가 요소들의 수치를 포함할 수 있도록 하였다. [그림 3-2]는 모든 항목을 나타낸 화면이고, [그림 3-10]은 모든 체크 버튼을 해제한 화면이다.



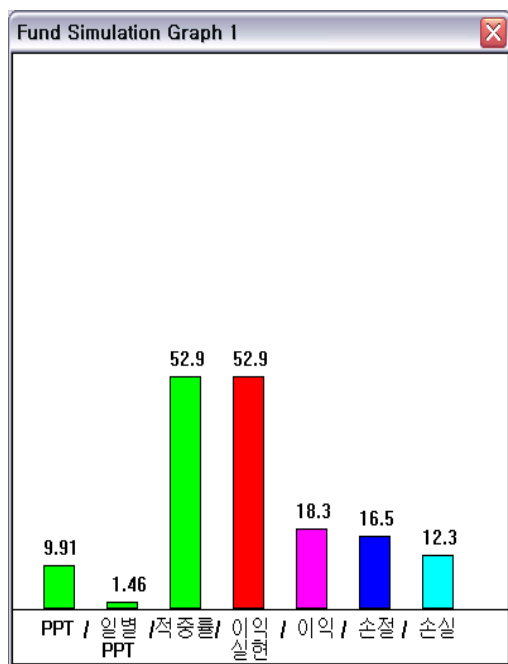
[그림 3-10] 모든 옵션 체크 버튼을 해제했을 때의 화면

이 중 ‘보유 일수’는 주가 추세 예측 엔진을 생성할 때 설정에 포함되는 것이고, 나머지는 앞의 Ⅲ-1절에 나온 시뮬레이션의 기존 성능 평가 척도이며 간략 그래프를 나타내는 요소이기도 하다.

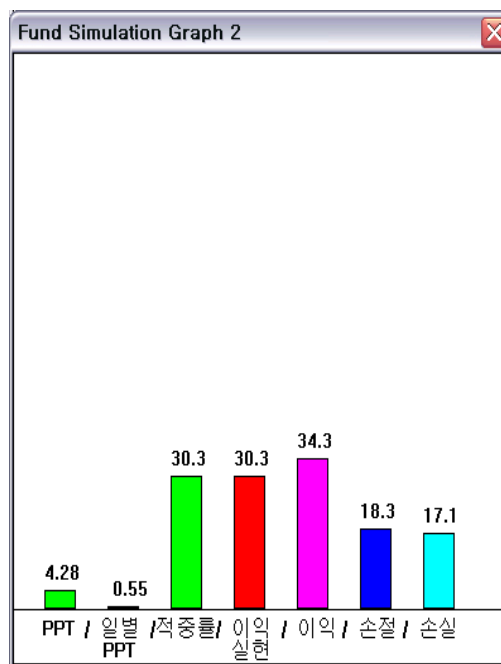
IV. 사례 분석

이 장에서는 본 논문에서 소개하는 시각화 도구를 통해 일반 투자자의 입장에서 두 개의 엔진 성능을 비교한다. 투자자가 두 엔진의 추천에 따라 거래를 하고 있다고 가정하고, 어느 엔진이 더 효용성이 있는지 알아본다.

A, B라는 엔진을 비교해 보기 위해 본 논문에서 제시하는 시각화 도구를 실행시켜 보았다. 먼저 간략 그래프를 보았는데 [그림 4-1], [그림 4-2]에서와 같이 PPT, 적중률 등의 기존 성능 평가 요소 면에서 엔진 A의 성능이 엔진 B의 성능보다 높은 것으로 나타났다.

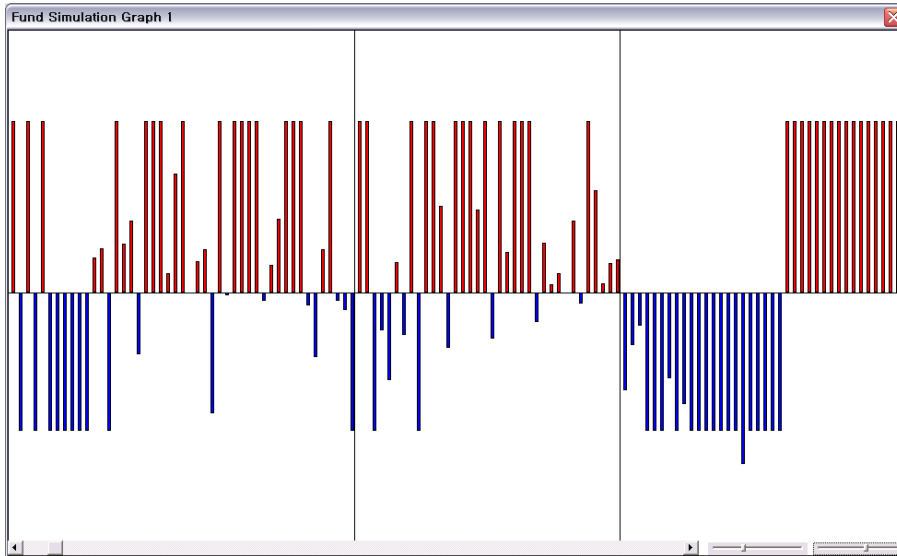


[그림 4-1] 엔진 A의 간략 그래프

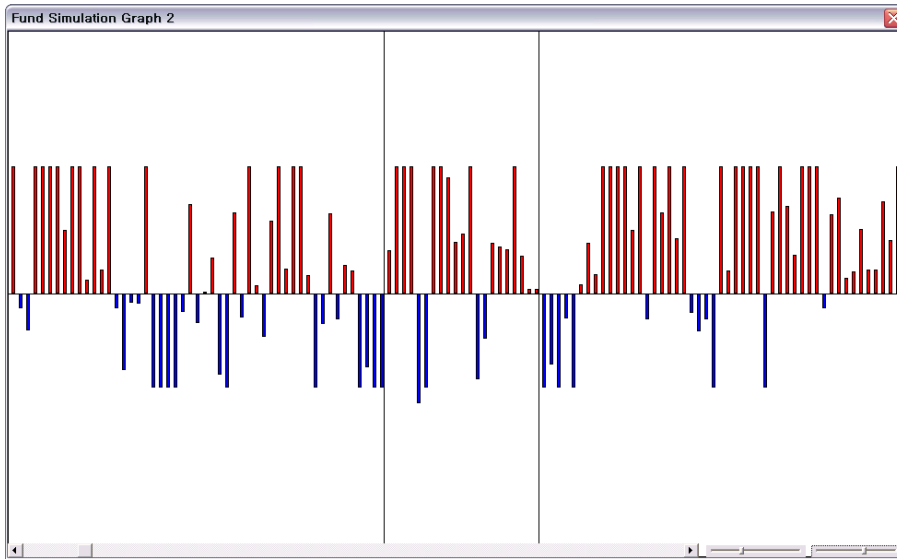


[그림 4-2] 엔진 B의 간략 그래프

그러나 더 자세한 거래 내역을 보기 위한 상세 그래프의 결과는 [그림 4-3], [그림 4-4]와 같다.

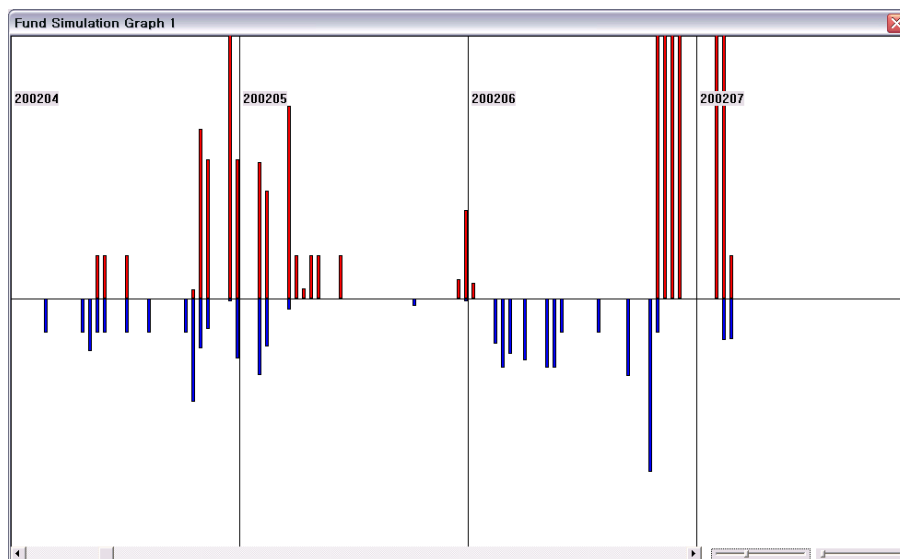


[그림 4-3] 엔진 A의 상세 그래프



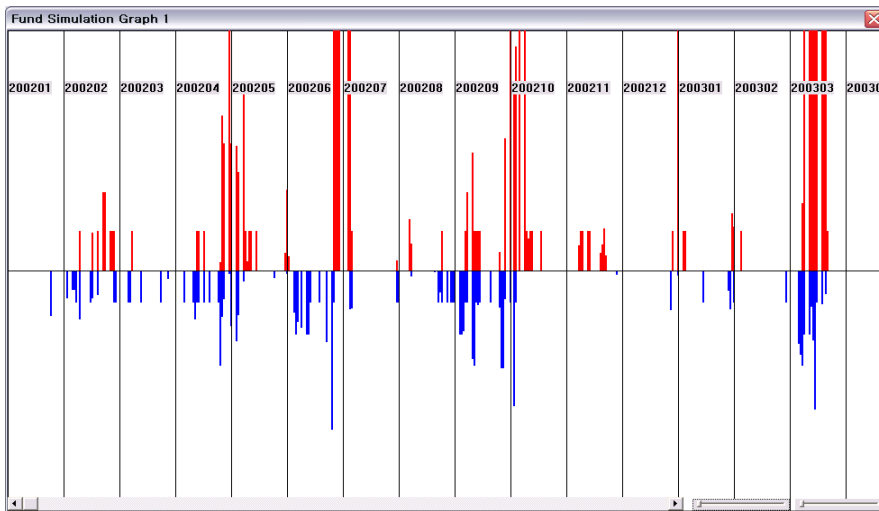
[그림 4-4] 엔진 B의 상세 그래프

위의 그림들로 보아 두 엔진 모두 매 달 끊이지 않고 일정 개수 이상의 추천을 낸 것으로 보인다. [그림 4-3]에서 엔진 A 그래프 막대의 길이인 이익 실현 폭이 [그림 4-4]에서의 엔진 B의 이익 실현 폭보다 크지만, 손실 횟수의 비율로 봤을 때 엔진 A가 엔진 B보다 현저히 많고, 또 이러한 손실 거래들이 한 구간에 몰려있는 것을 볼 수 있다. 연속되는 손실을 입은 후 다시 연속되는 이익을 달성할 수도 있지만 실제 투자자의 입장에서 엔진 A의 추천대로 거래를 수행하여 연이은 손실을 경험한다면 이 시점에서 거래를 중단할 가능성이 높고 차후에 이익을 달성할 것이라는 보장도 없으므로 계속 엔진 A를 신뢰하는 것은 불가능한 것으로 보인다. 따라서 총 거래를 기준으로 한 기존 성능 평가 요소인 PPT와 적중률 등의 수치는 엔진 A가 높지만, 일반 투자자라면 손실과 이익이 고르게 분포되어 있는 엔진 B를 더 선호할 것이다. 더 자세히 알아보기 위한 다른 그래프들은 다음과 같다.

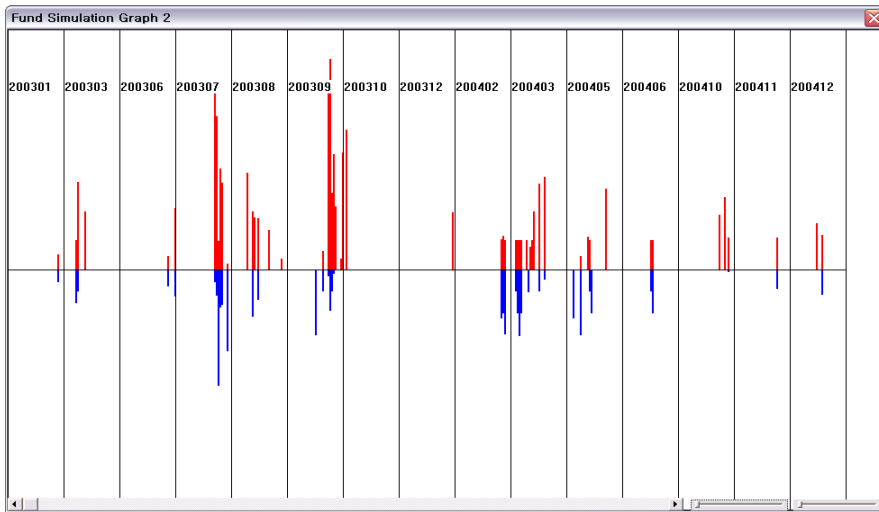


[그림 4-5] 엔진 A의 날짜별로 거래 정렬(누적 출력) 화면

[그림 4-5]는 엔진 A의 상세 그래프에서 문제가 되었던 손실이 집중되어 있는 달을 ‘날짜별로 거래 정렬(누적 출력)’한 것이다. 누적 출력의 경우 같은 날에 수행된 모든 거래의 수익률이 누적되기 때문에 막대의 길이가 매우 길어지므로 이 그래프는 슬라이더로 막대 길이를 가장 작게 설정한 것이다. 본 그래프를 통해 엔진 A의 경우 2002년 6월에 지속적으로 손실을 야기하는 추천을 여러 개 내었고 그달 말 나흘에 걸쳐 이익을 야기하는 추천을 많이 낸 현상을 발견할 수 있었다. 이러한 상황이 실제라고 가정한다면, 인내심이 아주 강한 투자자가 아닌 이상 6월 초부터 계속 투자 자금을 잃는다면 6월 중순쯤에 이르러서는 이후의 이익이 어찌되든 손을 털고 더 이상 엔진 A를 신뢰하지 않을 가능성이 클 것이다. 그러므로 손실을 야기하는 거래가 특정 기간에 집중되어 있는지의 여부는 투자자 입장에서 예측 엔진의 신뢰 여부 판단에 많은 영향을 끼치게 될 것이다. [그림 4-6]과 [그림 4-7]을 비교해 보면 엔진 A의 중복 추천수가 엔진 B의 그것보다 현저하게 많은 것을 알 수 있다.



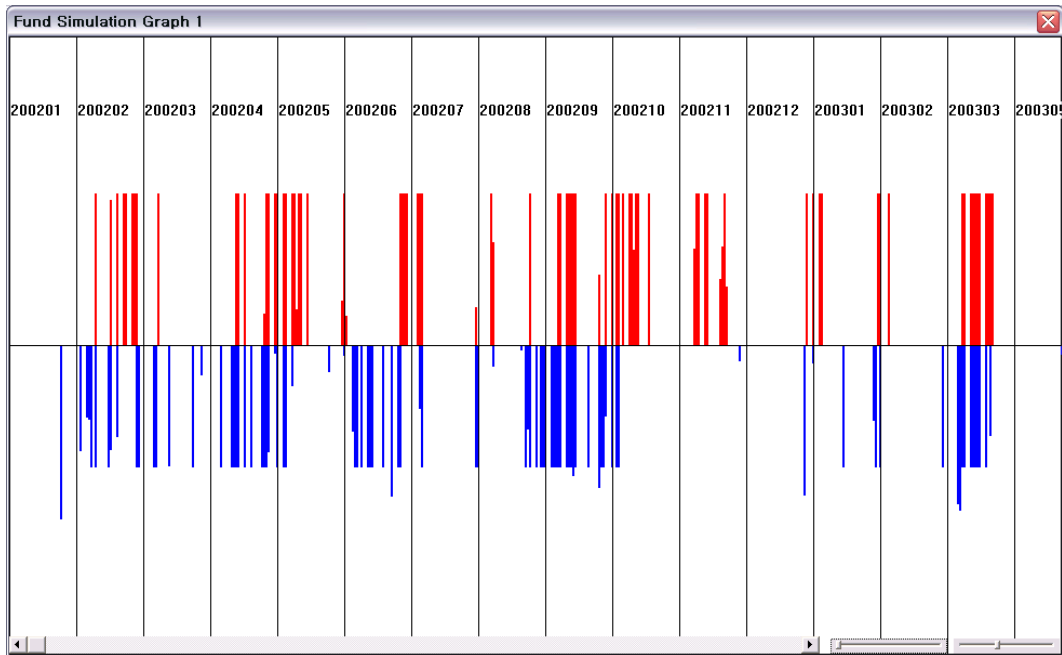
[그림 4-6] 엔진 A의 날짜별로 거래 정렬(누적 출력) 축소화면



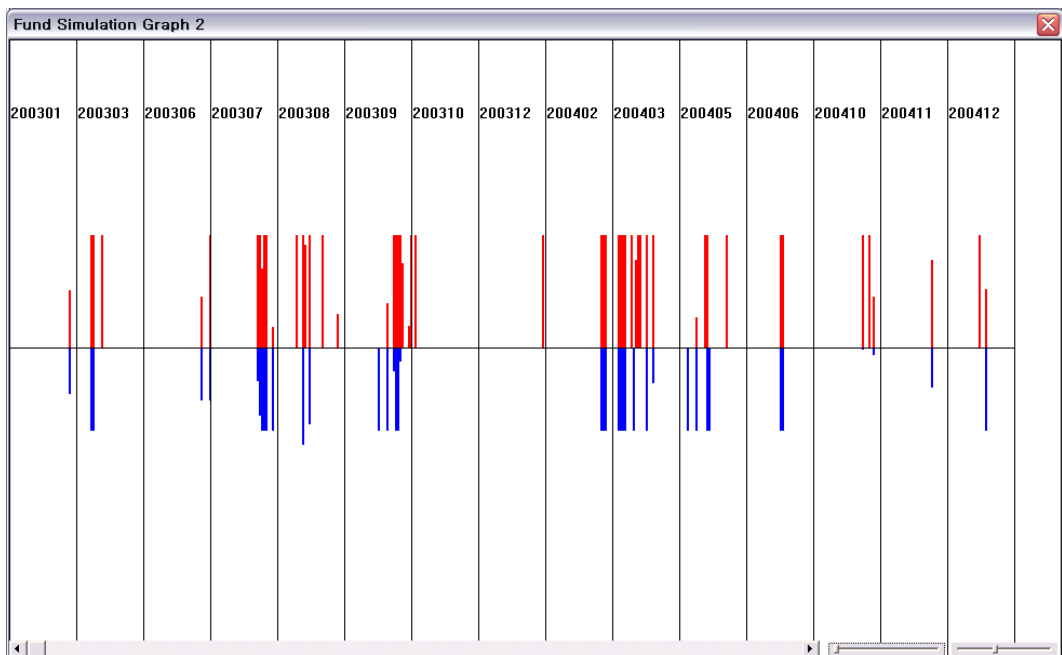
[그림 4-7] 엔진 B의 날짜별로 거래 정렬(누적 출력) 축소화면

[그림 4-6]의 경우, 투자 자금을 무한대로 가지고 있지 않은 투자자라면 이 모든 추천을 다 매수할 수 없기 때문에 엔진 A의 실제 투자 효율 가치는 떨어진다고 볼 수 있다. 반면, [그림 4-7]에서 보는 바와 같이 엔진 B의 중복 추천 수는 엔진 A의 중복 추천 수보다 적어 추천을 보다 안정적으로 생성한다는 것을 미루어 짐작할 수 있다.

이번에는 중복 추천일 때 누적을 제외하고 가장 큰 수익을 낸 거래와 가장 큰 손실을 낸 거래 값을 나타낸 ‘날짜별로 거래 정렬(중복 시 최대, 최소 출력)’ 그래프를 비교하여 보겠다. [그림 4-8]과 [그림 4-9]를 보면 중복 추천을 제거하고 난 후 엔진 A의 총 거래 분포 중 손실 거래의 비율이 엔진 B의 그것보다 많은 것을 발견할 수 있다. 이는 엔진 B의 추천 빈도가 비록 엔진 A의 추천 빈도에 비해 떨어지지만 적은 추천을 내더라도 안정적으로 이익을 내는 추천을 생성하는 경향을 보인다는 것을 확신할 수 있다.

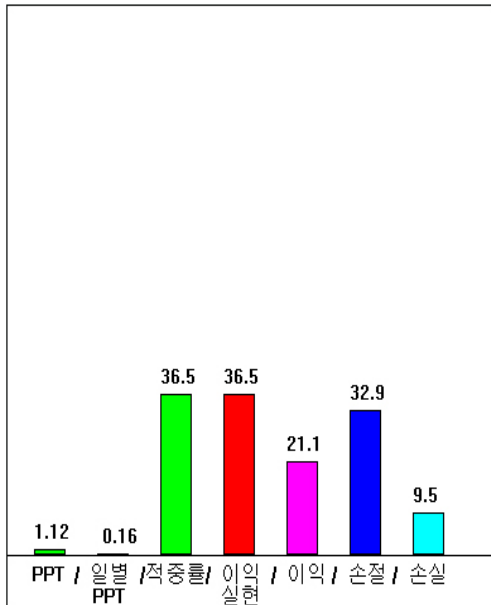


[그림 4-8] 엔진 A의 날짜별로 거래 정렬(최대, 최소 출력) 축소화면

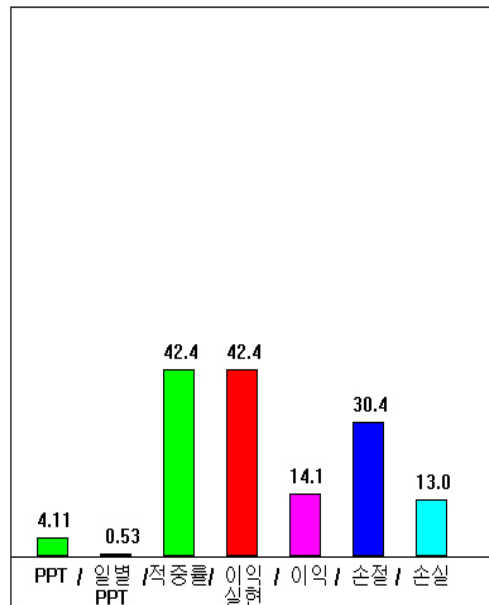


[그림 4-9] 엔진 B의 날짜별로 거래 정렬(최대, 최소 출력) 축소화면

마지막으로, 중복 추천일 때 누적을 제외하고 가장 큰 수익을 낸 거래와 가장 큰 손실을 낸 거래 값으로만 나타낸 결과를 보겠다. [그림 4-10]과 [그림 4-11]를 보면, 엔진 B의 성능이 더 좋다는 것을 확인할 수 있을 것이다. 이 두 그래프를 [그림 4-1], [그림 4-2]와 비교해 보아도, PPT와 적중률 등의 평가 요소 수치 면에서 중복을 제외한 결과가 엔진 A가 엔진 B보다 훨씬 큰 폭으로 떨어진 것을 확인할 수 있다. 이는 엔진 A가 기존 성능 평가 척도로는 성능이 높게 나왔지만 중복 추천으로 인한 수익률이 높았기 때문이었고, 이를 제외한 결과 기존보다 성능이 현저히 떨어져 효율성이 저하되는 추천을 많이 생성하는 경향이 있음을 알 수 있었다.



[그림 4-10] 엔진 A의 중복 추천을 제외한 결과



[그림 4-11] 엔진 B의 중복 추천을 제외한 결과

결론적으로, 실제 투자자 입장에서 엔진 A와 B를 비교했을 때 기존 성능

평가 척도 수치로 보면 엔진 A가 엔진 B보다 성능이 높은 것으로 나타났으나 손실 거래 집중도 등의 위험성, 같은 날의 중복 추천의 비율 등으로 보아 투자자에게 선호되는 예측 엔진은 엔진 A보다 엔진 B가 될 것이라고 추정된다.

위의 결과와 같이 PPT, 일별 PPT, 적중률 등의 기존 성능 평가 척도로만으로는 실제 투자자가 예측 엔진을 선택하여 이의 추천에 따라 거래를 하기에는 부족한 면이 있다. 따라서 본 논문에서 제시한 시각화 도구를 통하여 다각적으로 분석한 후, 더 효용성 있는 예측 엔진을 선택하는 것이 실제 투자에 유익한 선택이 될 것으로 보인다.

V. 결론

본 논문에서는 신경망 기반 주가 예측 시스템 도구의 분석과 성능 평가를 위한 펀드 시뮬레이션 결과 파일의 그래프를 통해서 보다 실용적이고 실제 투자자의 성향에 맞춘 성능 평가 분석을 시도하였다.

기존의 통합 시뮬레이션 거래 시스템의 성능 평가에서는 PPT, 일별 PPT, 적중률, 이익실현·이익·손실·손절 비율 등이 기준 요소로 사용되었으나, 이 논문에서는 기존의 수치화된 값으로만 비교할 수 있었던 것에서 벗어나 직접 그래프로 보여주어 사용자의 이해가 쉽도록 하였다. 또한 여러 가지 모드의 그래프를 통하여 실전 거래에 있어서 주가 추세 예측 엔진의 선택에 중요한 요소가 될 다른 평가 요소들을 접목시켰다.

이를 입증하기 위해서 시뮬레이션을 통한 사례 분석을 하였다. 두 가지 엔진의 비교 결과, 기존의 성능 평가 요소 수치가 높은 엔진의 경우 손실 거래가 특정 기간에 집중되어 있었고 높은 수익률을 나타내었던 거래 또한 다른 특정 기간에 집중되어 있는 현상을 발견할 수 있었다. 이 현상을 통해 기존 성능 평가 요소만으로는 실제 투자자가 예측 엔진의 성능을 평가하기에는 미흡한 점이 있었다는 것을 확인할 수 있었다. 그러나 오히려 기존 성능 평가 요소 수치가 낮은 엔진의 경우 고른 분포의 수익성 있는 거래와 특정 기간에 집중되어 있지 않은 손실 거래 등으로 실제 투자를 수행하기에는 안정성에 있어 더 적합하다는 것을 알 수 있었다. 이러한 결과들로 미루어 보아, 평가 요소 수치만을 보고 성능을 평가하는 것보다, 다각적인 분석을 통하여 주가 추세 예측 엔진을 선택하는 것이 실제 투자자를 만족시키는 데

에 보다 효율적이라고 판단된다.

본 논문의 사례 분석을 진행하기 위해 여러 가지 엔진들을 비교, 분석한 결과 특정 기간에 성능이 매우 좋았던 엔진이 또 다른 특정 기간에는 성능이 부진하여 결과적으로는 전반적 성능이 저하될 수 있음을 확인하였다. 본 논문에 제시된 모듈을 사용하여 여러 엔진을 다각도로 비교, 분석하는 것뿐만 아니라 더 좋은 성능의 주가 추세 예측 시스템 개발을 위해 특정 시기에 성능이 좋은 엔진들을 종합적으로 선택하여 하나의 엔진으로 결합, 지속적으로 성능이 좋은 엔진을 탄생시킬 수 있는 연구가 추가적으로 이루어져야 할 것으로 보인다.

참고문헌

- [1] A. Refenes, "Neural Networks in the Capital Markets", John Wiley and Sons, 1995
- [2] 문정희, "기계학습 기반의 주가시계열 예측에서의 성능 평가 연구 사례", 2005
- [3] J. O., J. W. Lee, J. Lee, and B. Chae, "Dynamic Asset Allocation Exploiting Predictors in Reinforcement Learning Framework", European Conference on Machine Learning, pp. 298~309, 2004
- [4] J. W. Lee, S. Kim, J. Lee, and Chae, "An Intelligent Stock Trading System based on Reinforcement Learning", IEICE Trans. on Information and Systems, 2003.
- [5] J. W. Lee and J. O, "A Multi-agent Q-learning Framework for Optimizing Stock Trading Systems", In Proceedings of International Conference on Database and Expert Systems Applications, pp. 153-162, 2002.
- [6] 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.
- [7] 정의석, "정의석의 주가학 원론 - 과감하게 투자해야할 때 냉정하게 물러서야할 때", 청림출판, 2003
- [8] T. Hellström, "A Random Walk through the Stock Market", ph.D. Thesis, Department of Computing Science, Umeå University, 1998.

- [9] T. Hellström and K. Holmström, "Predicting the Stock Market",
Technical Report IMA-TOM-1997-07, Department of Mathematics and
Physics, Mälardalen University, Sweden, 1998
- [10] 김현진, "신경망 기반 주가 예측 시스템의 성능 평가 요소 분석", 2005

ABSTRACT

Implementation of a Visualization Tool for Performance Evaluation of Neural Network based Stock Price Forecasting System

Chung, Seung Hye

Major in Computer Science

Graduate School

Sungshin Women's University

The stock market is the core of the modern economics in capitalism. So the studies on predicting the stock prices are advanced in many research areas such as business economics, economic science, mathematics, physics and so on. Especially studies in computer science primarily use automatic learning methods such as neural networks. However, those studies are not focused on whether their systems can show the performance reported when actually applied to the real stock market. Therefore more objective and systemic evaluation criterion, considering realistic constraints, is needed to get meaningful research results of stock market prediction.

In this thesis, a visualization tool for a new performance evaluation scheme is introduced. This tool provides various visual materials related to prediction performance. The new evaluation scheme in 'NNShell', an integrated tool for stock trading system, is different from those in the previous works, in that it considers more realistic factors for evaluation of prediction performance and the proposed visualization tool is to be integrated into 'NNShell'.

In the case study, the performance evaluation using the proposed visualization tool is compared with that using existing performance metrics. This case study shows that many-sided analysis of prediction performance is possible using this visualization tool. Also, detailed comparison of trading logs including profit rate, trading frequency and distribution, is performed for several distinct prediction engines.