

# 의사결정 트리를 이용한 학습 에이전트 단기주가예측 시스템 개발

## - A Development for Short-term Stock Forecasting on Learning Agent System using Decision Tree Algorithm

서 장 훈 \*

Seo Jang Hoon

장 현 수 \*\*

Chang Hyeon Soo

### Abstract

The basis of cyber trading has been sufficiently developed with innovative advancement of Internet Technology and the tendency of stock market investment has changed from long-term investment, which estimates the value of enterprises, to short-term investment, which focuses on getting short-term stock trading margin. Hence, this research shows a Short-term Stock Price Forecasting System on Learning Agent System using DTA(Decision Tree Algorithm) ; it collects real-time information of interest and favorite issues using Agent Technology through the Internet, and forms a decision tree, and creates a Rule-Base Database. Through this procedure the Short-term Stock Price Forecasting System provides customers with the prediction of the fluctuation of stock prices for each issue in near future and a point of sales and purchases.

A Human being has the limitation of analytic ability and so through taking a look into and analyzing the fluctuation of stock prices, the Agent enables man to trace out the external factors of fluctuation of stock market on real-time. Therefore, we can check out the ups and downs of several issues at the same time and figure out the relationship and interrelation among many issues using the Agent. The SPFA (Stock Price Forecasting System) has such basic four phases as Data Collection, Data Processing, Learning, and Forecasting and Feedback.

**keyword : Short-term Stock Forecasting, Decision Tree Algorithm**

\* 명지대학교 산업시스템공학부

\*\* 경기공업대 산업경영시스템과

It is also important to set a proper forecasting time interval for accuracy of the Agent forecasting. For the sake of it, this research will focus on discussing the derivation of sufficient forecasting time interval for each issue; it has been accomplished by testing the same data with several different forecasting time intervals.

## 1. 서 론

### 1.1 연구배경 및 목적

오늘날 주식투자의 경향은 인터넷 기술의 발달로 인한 사이버 트레이딩 기반의 발전과 정보취득의 용이성으로 인하여 기업의 가치를 평가한 장기적인 투자에서 단기적인 투자로 매매이익을 실현하는 방향으로 바뀌었다. 결과적으로 과거에는 통합형 Assistant Agent였다면, 현재에는 웹상에서의 지능형 에이전트, 미래에는 전자상거래를 위한 에이전트로 변화해갈 것으로 예측된다. 이러한 환경의 변화로 인하여 기존의 주가예측에 쓰이는 기본적 분석(기업의 내재가치 분석)과 기술적 분석(주가의 추세분석)은 단기투자에서의 의사결정에는 별다른 도움을 주지 못하는 현실이다. 따라서 단기주식투자를 위하여 적절한 의사결정에 도움을 줄 수 있는 방안이 요구되고 있다. 인터넷 기술의 발달로 사이버 트레이딩의 기반이 충분히 발전하였고, 주식투자의 경향이 기업의 가치를 평가한 장기적인 투자에서 단기적인 투자로 매매이익을 실현하는 방향으로 바뀌었다. 이에 본 연구에서는 단기주식투자를 위하여 에이전트(Agent) 기술을 이용하여 인터넷에서 실시간으로 관심종목의 주가를 수집해서, 그 정보를 바탕으로 의사결정 트리를 생성하여 규칙 데이터베이스를 만들어 각 종목별로 가까운 미래의 주가의 등락을 예측하여 사용자에게 매도/매수시점을 제시할 수 있는 단기주가예측 에이전트 시스템을 개발을 하여 사례연구로서 제시한다.[1]

본 연구에서 제시하는 SFA(Stock Forecasting Agent) 시스템은 자료수집단계, 자료가공단계, 학습단계, 예측 및 피드백 단계의 4단계로 수행된다. 또한 에이전트 예측의 정확도를 위하여 예측시간간격의 설정이 중요한데, 이를 위하여 동일 데이터를 여러 예측시간간격으로 나누어서 예측의 정확도를 높일 수 있는 각 종목별 예측시간간격을 구하는 것에 관하여 논의한다.

### 1.2 연구 방법

과거부터 주가를 예측하려는 노력과 관련 연구는 계속 되어왔다. 최근 연구로 (임도형, 이일병, 2000) 신경망을 이용한 매도/매수 주식 종목을 선정하는 연구에서는 주가를 시계열 데이터의 일종으로 보고 주가를 신호와 잡음이 혼합된 것으로 가정하여 주가의 변동에 영향을 주는 많은 변수들의 특성을 고려하여, 전 종목에 대한 등락을 예측하지 않고, 예측율이 높은 종목을 선정하는 것을 목표로 하고 있다.

그리고, (오성민, 김성집, 2000) 다른 연구에서는 주식의 상한가시 잔량과 일일거래량의 관계를 통하여 주가의 단기예측을 하려는 노력을 하고 있다. 이 연구에서는 기술적 분석의 많은 요인들 중에서 기존에 많이 연구해보지 못한 시계열 적인 인자를 가지고 단기간의 주가를 예측하려고 하였다. 즉, 주식이 상한가에 도달하였을 경우 그 상한가 격의 잔량과 그 주식의 일일거래량을 비교하여 그 두 관계가 다음날 주가에 어느 정도의 영향을 미치는지 회귀분석을 통하여 상관성을 분석하고 통계적 자료를 토대로 단기간의 주가를 상한 잔량 대비 일일거래량에 비추어 의사결정 지표로 제시하였다. 이렇듯 여러 방식으로 주가예측에 관한 연구의 접근이 이루어지고 있지만, 실제 실시간으로 이루어지는 주식투자에 응용하고 투자자의 의사결정에 도움을 줄 수 있는 시스템의 개발이 필요하다.[2]

따라서 본 연구에서는 에이전트 기술을 이용하여 실시간으로 단기간의 주가를 예측하여 투자자의 의사결정에 참고자료를 제시하였다. 이러한 지능화된 에이전트의 학습방법에는 최근 들어 각광받는 데이터 마이닝 기법중의 하나인 의사결정 트리 알고리즘을 사용한 지능화된 에이전트(Agent) 기술을 이용하여 인터넷에서 실시간으로 관심종목들의 주가정보들을 수집해와서, 그 정보를 의사결정 트리 학습(Decision Tree Learning Algorithm)을 이용하여 각 종목별로 에이전트가 가까운 미래의 주가의 등락을 예측할 수 있는 단기주가예측 에이전트 시스템에 관하여 논의하고자 한다.

## 2. 이론적 고찰

### 2.1 에이전트의 정의

에이전트란 사전적 의미로 대행자, 대리인이란 뜻이며, 컴퓨터 분야에서는 작업을 대행해주는 프로그램이라고 생각하면 된다. 하지만 컴퓨터 분야에 좀더 애착이 있는 사람들은 보다 중요한 의미를 부여한다. 에이전트에 관한 연구는 연구하는 사람들마다 다르게 정의하고 있으며, 어떤 경우에는 기존의 정의 자체를 바꾸기도 한다. 현재 에이전트에 대해 가장 많이 연구한 인공지능 분야의 사람들은 에이전트를 분산 환경에서 작업을 수행하는 지적인 특성을 갖는 응용 프로그램이라고 말한다. 또 사회과학적인 배경을 갖고 있는 사람은 하나의 일을 상호 협력해 해결하는 응용 프로그램으로 정의하고, 휴먼 컴퓨터 인터페이스(HCI)를 공부하는 사람은 '자원이 생각을 하면서 사람이 요구하는 사항을 처리해주는 프로그램'으로 생각한다.

이와 같이 에이전트는 연구하는 사람들마다 서로 다르게 정의하기 때문에 에이전트를 한마디로 설명하기는 어려운 일이다. 그러나 본 논문에서 논의되는 에이전트는 기본적으로 지능형 소프트웨어 에이전트로써 큰 범주로 보면 일종의 프로그램이라 볼 수 있다. 물론, 로봇과 같은 하드웨어 에이전트도 에이전트의 범주에 포함시키기도 하지만 여기서는 소프트웨어 에이전트에 대해서만 다룬다. 하지만 이때 대두되는 질문은 과연 소프트웨어 에이전트라는 것이 기존의 일반 프로그램과 무엇이 다르냐 하는 것이다. 바꾸어 표현한다면 기존의 소프트웨어, 예를 들어 워드 프로세서, 스프레드시트, 게임

들도 모두 에이전트라고 할 수도 있지 않느냐 하는 것들이다. 이를 위해서 단순한 프로그램이 아닌 소프트웨어 에이전트이기 위해서는 어떤 기능을 수행하고 어떤 역할을 담당해야 하는가에 대한 답변이 요구되는데 이에 대한 답변이 바로 에이전트의 정의가 될 수 있다.[4]

이러한 에이전트의 정의를 전자우편 에이전트를 예로 들어 설명할 수 있다. 전자우편 에이전트의 목적은 사용자가 수작업으로 해왔던 전자우편의 분류, 검색, 삭제, 전달 등의 작업을 대신해주는 것이며, 특성상 Unix나 Windows 등의 운영체제 환경 하에서 동작한다. 지식의 처리를 위해서 사용자의 요구사항이 조건-작용 규칙 (condition-action rule) 의 형태로 지식베이스에 저장되고, 이 규칙의 조건부분은 추론 메카니즘을 통해 전달된 메시지와 매치가 되는지 결정된다. 가령 'security'에 관한 전자우편이 도착했을 때 사용자가 현재 있는 곳으로 전달해 달라는 규칙이 있을 때, 'security alert'라는 제목의 전자우편이 오면 조건이 만족되었으므로 사용자의 위치를 파악하여 전화나 팩스 등의 방법으로 메시지를 전달하게 된다. 전자우편 에이전트는 한 번의 수행으로 종료되는 것이 아니고 데몬(daemon)처럼 항상 동작되면서 새로운 전자우편 메시지를 모니터링하게 되며, 전화번호 데이터베이스나 사용자의 일정 등과 같은 시스템의 다른 자원과도 메시지 교환을 통해 협력하게 된다.

에이전트에 대한 정의는 절대적인 것은 아니지만 어떤 소프트웨어가 에이전트인지 아닌지를 나름대로 판단하는데 좋은 지표가 될 수 있다. 물론 판단대상이 에이전트다 또는 아니다라고 단정짓는 것은 좋은 접근 방향이 아닌 것 같고, 다만 위에서 정의된 항목에 얼마나 많이 부합되는가에 따라 에이전트의 역할을 좀 더 충실히 한다고 말할 수 있겠다.[3][5]

## 2.2 학습 에이전트

학습 에이전트(Learning Agent)는 사용자의 웹 상에서의 활동을 관찰하고 사용자가 어떤 내용에 관심을 가지고 있는지 판단하여 사용자에게 알맞은 내용을 전달하도록 하는 것을 말한다. 사용자의 웹 내에서의 행동 중에서 중요하게 사용되는 것은 특정한 페이지를 보는 시간, 인쇄한 페이지, 전자 상거래 사이트의 경우에는 구매한 상품과 쇼핑 카트에 넣은 상품 등을 들 수가 있으며, 이러한 사용자의 행동을 관찰하기 전에 선행되어야 할 것은 웹 페이지의 내용들에 대한 정보가 일정한 기준에 따라 분류되어 있어야 한다는 것이다. 즉, 사용자에게 제공될 자료들에 대한 정보(meta-data)가 제공될 내용과 함께 데이터베이스화 되어 있어야 한다는 것이다. 구축된 내용 데이터베이스와 관찰된 사용자의 웹 사용 습관 토대로 데이터 마이닝의 과정을 거쳐 사용자의 성향과 관심이 결정되고 사용자에게 알맞은 내용이 제공된다. 이러한 과정은 시스템이 사용자의 행동을 바탕으로 사용자의 성향을 학습한다는 점에서 학습 에이전트라고 불리고 있다. 현재 학습 에이전트 기술은 사용자가 보는 페이지를 분석하는 수준을 뛰어넘어 어떤 문장을 읽는지를 분석하는 수준까지 개발되고 있다. 일반적으로 협업 필터링이 효력을 발휘하려면 일정 수 이상의 사용자가 필요한 것에

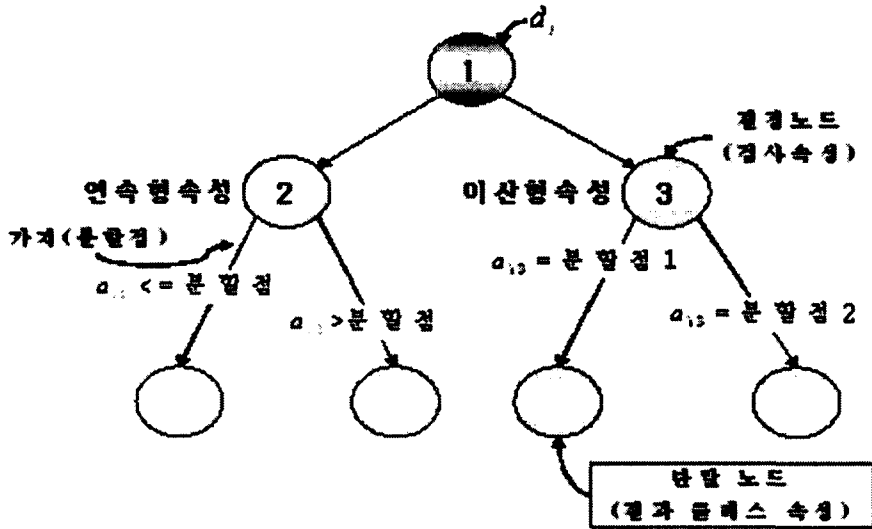
비해 학습 에이전트는 분석에 사용되는 자료가 다른 사용자 자료와의 비교를 필요로 하지 않기 때문에 사용자가 적은 숫자인 경우에도 적절한 내용을 전달할 수 있다. 그러나 협업 필터링이 사용자가 자신의 선호도에 관한 내용을 입력하는 즉시 사용자들에게 맞는 내용을 전달할 수 있는 데 비해 학습 에이전트 방식은 사용자의 웹에서의 행태를 일정 시간이상 관찰한 이후에야 정보 제공이 가능하다는 점에서 차이가 있다. 학습 에이전트 방법을 실행하는 데 가장 어려운 점은 현재의 사이트의 내용을 학습 에이전트를 사용할 수 있는 환경으로 재조정해야 한다는 데 있다. 또한 쿠키를 활용하거나 회원 등록을 통해 사용자를 확인하는 과정이 필수적이라는 점도 어려운 점이 될 수 있다. 그것은 사용자들이 여러 번 접속하거나 긴 시간동안 사이트를 사용한 후에야 사용자들에게 의미가 있는 서비스를 제공할 수 있기 때문이다. 그러나 일단 사용자에게 적절한 내용을 제공할 수 있는 단계를 넘어서면 사용자 정보가 계속 축적되어서 더욱 효과적으로 사용자에게 알맞은 정보를 제공할 수 있다.

적용 사이트는 대중을 상대로 한 사이트 보다는 사용자의 숫자가 많지 않더라도 주기적인 방문을 기대할 수 있는 커뮤니티나 뉴스 제공 사이트에 적합하다. 또한 사용자 확인만 되면 사용자의 신상정보를 공개할 필요가 없이 개인화 하는 것이 가능하게 되기 때문에 최근에 관심의 초점이 되고 있는 사생활 보호라는 측면에서도 바람직하다고 할 수 있다.

## 2.3 의사결정 트리

의사결정 트리(Decision Tree)는 불확실한 사상(思想)에 직면한 의사결정에 있어서 형성되는 분기점의 동형이라고 정의내릴 수 있으며, 의사결정 트리의 각 비 단말 노드(nonterminal node)에는 트리의 단말 노드에 저장된 개체 등에 관한 질문이 들어있다. 질문에 대한 대답에 의해서 그 질문이 소속된 노드의 자식 노드로 연결된 가치를 따라 밑으로 내려가면서 트리의 어느 위치에 새로운 개체를 저장할 것인지 결정하게 된다. 의사결정 트리의 뿌리(root)로부터 다른 노드로의 경로는 일련의 질문과 해답에 대응이 된다. 단말 노드에 저장된 개체들은 모두 다 뿌리로부터 각 노드로 가는 경로 위에 있는 질문과 해답들과 일치하는 개체들이다. 여기서 질문은 속성(Attribute)에 해당되고 해답은 그 속성의 값, 즉 부류(Class)에 해당된다. 따라서 뿌리로부터 노드의 경로에 있는 질문들과 해답들은 모든 가능한 예제들의 집합의 어떤 부분 집합을 정의하는 특징들의 논리곱형태로 표현된다. 이에 대하여 < 그림 1 >에서 나타내었다.

결정 트리를 구현하기 위해 트리의 노드에 대한 추상적 자료 타입을 정의해야 한다. 결정 트리는 속성, 예제들의 리스트, 자식 노드의 리스트로 구성되어 있다. 결정 트리를 찾아내는 방법은 뿌리로부터 아래로 결정 트리를 만들어 가는 것이다. 뿌리는 항상 모든 훈련예제들의 집합을 포함하고 있다. 다른 모든 노드들은 뿌리로부터 노드로의 경로에 의해 결정된 예제들의 부분집합을 포함하고 있다. 한 노드에 속한 모든 예제가 하나의 부류(Class)로 결정되면 그 노드는 단말 노드가 된다. 단말노드의 경우에는 분류를 위하여 더 이상 식별을 할 필요가 없다.[6]



< 그림 1 > 의사결정 트리

본 연구에서 에이전트의 학습에 의사결정 트리 생성에 < 그림 2 >에서와 같이 C4.5 알고리즘을 사용한다. 데이터 마이닝을 위해 사용되는 의사결정 트리의 구축에 관련된 대표적인 연구중 하나인 C4.5는 ID3 알고리즘을 확장한 것으로 기존의 ID3 알고리즘은 이산적인 속성값만 처리할 수 있는데 반해 이 알고리즘은 연속적인 속성값도 처리할 수 있다.

$$INFO(S) = - \sum_{j=1}^k P_j \times \log_2(P_j) \text{ bits}$$

where,  $INFO(S)$  is entropy

$$P_j = \frac{freq(C_j, S)}{|S|}$$

$freq(C_j, S)$ : 데이터 set  $S$ 가 Class  $C_j$ 에 속한 빈도수

< 그림 2 > C4.5의 엔트로피 함수

C4.5는 J Ross Quinlan이 의사결정 트리 학습을 구현한 프로그램으로, 다층 퍼셉트론 신경망과 비슷한 성능을 보이며, 분류 문제를 해결해 주는 분류기(classifier)이다[10]. C4.5는 엔트로피(entropy)를 이용하여 이진 결정 트리를 만들어가면서 분류 문제를 해결한다. 결정 트리는 분류에 영향을 미치는 변수가 내부 노드(internal node)로 설정되

며, 그 값에 따라 데이터를 분류한다. 따라서, 최종 결정된 트리의 루트(root)로부터 리프 노드(leaf node)까지의 경로(path)상의 변수 값의 범위가 해당 클래스(class)를 분류하는 조건이 된다. C4.5는 결정 노드에서 분할 기준이 되는 최적 검사 속성을 선택하기 위해서 속성 선택 척도로 정보획득량(Information Gain)을 사용한다. 이 척도는 정보 이론의 엔트로피(Entropy) 개념을 사용하는데 엔트로피 값은 데이터의 분할이 잘 될수록 작은 값을 갖게 된다. 따라서 검사 속성 후보 중에서 엔트로피 값이 최소가 되는 속성을 해당 결정 노드의 검사 속성으로 선택하게 된다. C4.5에서 결정 트리를 만들 때 사용되는 엔트로피 함수는 < 그림 2 >과 같으며, 이 엔트로피 함수를 이용하여 엔트로피 값이 작아지는 방향으로 내부 노드 변수를 선정한다.[9]

## 2.4 주가 변동 요인에 대한 가정

주가는 매수 세력과 매도 세력에 의해서 변동폭이 결정되는 시계열 데이터이다. 매수 세력과 매도세력은 과거의 주가 뿐 아니라 환율이나 금리, 혹은 각 종목에 대한 특정 뉴스에 의하여 영향을 받는다. 이처럼 주가란 수많은 변수들의 영향을 받기 때문에 정확한 예측은 거의 불가능하다고 볼 수 있다. 그러나 많은 변수들 중에서 주가에 많은 영향을 주는 중요한 변수들을 가려낼 수 있고, 그 변수들간의 관계와 규칙을 찾아낼 수 있다면 어느 정도의 오차범위에서 주가를 예측할 수 있을 것이다.

본 연구는 다음과 같은 가정에서 출발한다.

가정 1 : 각 종목별로 주가변동의 형태가 다르다. 즉, 각 종목의 주가에 영향을 주는 변수들의 영향 정도가 다르다.

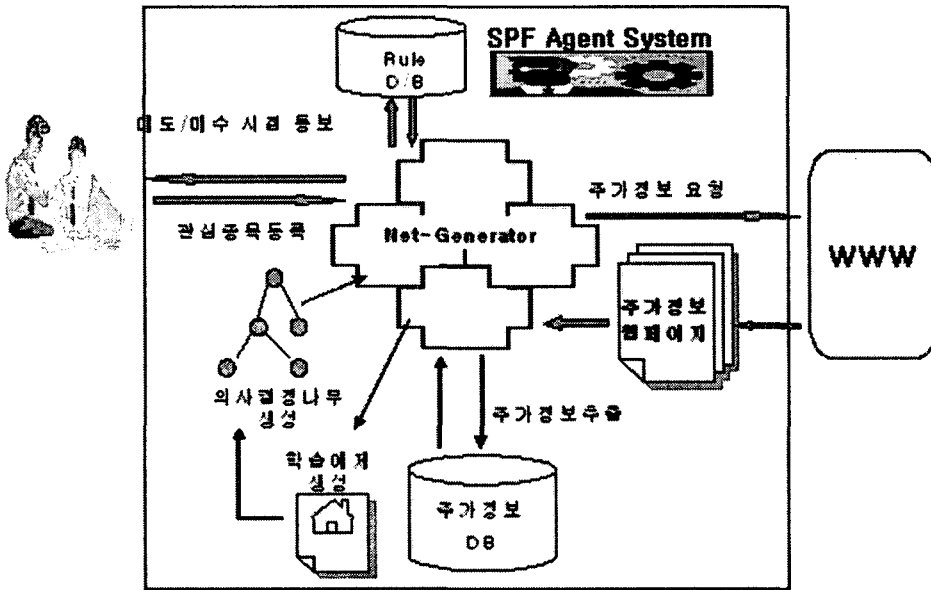
가정 2 : 주가에 영향을 미치는 요인들은 주가에 독립적으로 영향을 주는 것이 아니라 한 요인에 변동이 생기면 곧 다른 요인들에게도 영향을 준다. 즉, 어떤 종목에 호재 또는 악재가 있으면 거래량이 증가 또는 감소하거나 주가가 상승 또는 하락한다. 따라서 거래량과 매도/매수의 관계를 관찰 학습하는 것은 다른 요인들의 영향도 같이 고려하는 것이다.

## 3. 단기주가예측 에이전트 시스템 개발

### 3.1 단기주가예측 에이전트 시스템 구조

아래 < 그림 3 >은 단기주가예측 에이전트 시스템의 전체 구조를 나타내고 있다. 먼저 사용자의 관심종목을 입력으로 받아서 해당종목의 주가정보 웹 페이지를 인터넷을 이용하여 수집해 오고, 수집된 웹 페이지를 분석하여 필요한 정보만 데이터베이스에 저장을 한다. 이렇게 축적된 데이터는 의사결정 트리를 생성하기 위한 학습예제데이터로 변환되어 의사결정 트리를 생성하고, 생성된 의사결정 트리를 규칙으로 변환시

켜 데이터베이스에 저장한 후 새로운 데이터를 입력받으면 기존의 규칙들을 검색하여 앞으로의 주가의 추세를 예측하여 사용자에게 매도/매수 시점을 통보한다. 이러한 활동은 반복적으로 수행되어 기존의 예측의 오류여부를 확인할 수 있고, 다시 새로운 의사결정 트리를 생성한다.



< 그림 3 > SPF Agent System 구조

본 연구에서 제시하는 단기주가예측 에이전트 시스템은 여러 기술들을 복합적으로 사용하고 있다. 먼저, 인터넷을 이용하여 정보를 가지고 오는 기술이 필요하고, 이를 분석하여 데이터베이스에 필요한 정보를 저장하는 기술, 또한 저장된 정보를 바탕으로 의사결정 트리를 구성하는 기술과 새로 입력받은 정보를 의사결정 트리를 이용하여 예측할 수 있는 기술이 필요하다.[10]

### 3.2 단기주가예측 에이전트 수행절차

위에서 제시한 SPF Agent System 구조에 따른 에이전트의 수행절차는 연속적이고 반복적인 4개의 Backbone 활동단계로 나누어 정리하면 다음과 같다. 그리고 본 논문에서는 4단계 수행절차에 따른 필요한 소스를 사례와 함께 자료로서 제시하였다.

- (1) 자료 수집 단계 : 인터넷상의 정보를 수집(Fetch)하고, 분류(Parsing)해서 정리한 후에 데이터베이스에 저장한다.
- (2) 자료 가공 단계 : 데이터를 가공하여 학습예제(Training Example)를 생성한다.



- (3) 학습 단계 : 학습예제를 이용하여 의사결정 트리를 생성하고, 생성된 의사결정 트리를 규칙(Rule)으로 변환하여 데이터베이스에 저장한다.
- (4) 예측 및 피드백 단계: 새로 입력받은 현재의 추가정보를 이용하여 기존에 생성된 규칙 데이터베이스를 검색하여 어떠한 규칙에 속하는지 예측하여 사용자에게 통보한 후, 앞의 단계를 반복한다.

아래 < 그림 4 >은 에이전트 프로그램의 초기화면이다. 여기서는 추가정보를 제공하는 웹사이트 주소와 관심 종목명을 입력하는 창과 그 종목의 추가정보를 저장하고 있는 데이터베이스의 정보를 보여주는 창으로 크게 나눌 수 있다.

**SPF Agent System**

사이트 주소  기본 메뉴  
 관심종목명  1. 추가정보 가져오기  
 거래날짜와 시간 2. 학습예제 만들기  
 Fetching time : 오후 2:12:05 3. 학습시작  
 총 데이터 갯수 3895

날짜	시간	현재가	상락	전일대비	매도호가	매수호가	체결량
2003-05-15	오전 09:00	46900		1050	0	0	10
2003-05-15	오전 09:01	46900		1050	0	0	2050
2003-05-15	오전 09:02	45700	▼	800	45700	45200	30
2003-05-15	오전 09:02	45700	▼	850	45700	45200	40
2003-05-15	오전 09:02	46900	▼	1000	46900	45950	20
2003-05-15	오전 09:03	46750	▼	750	46900	45950	50
2003-05-15	오전 09:03	46750	▼	800	46900	45950	60
2003-05-15	오전 09:03	46750	▼	750	46900	45950	1520
2003-05-15	오전 09:03	46950	▲	800	46900	45950	400
2003-05-15	오전 09:03	46950	▲	750	46900	45950	1200
2003-05-15	오전 09:03	46950	▲	900	46900	45950	1200
2003-05-15	오전 09:03	46950	▲	750	46900	45950	1524
2003-05-15	오전 09:03	47750	▲	1000	47900	46500	3520
2003-05-15	오전 09:03	47750	▲	1100	47900	46500	1100
2003-05-15	오전 09:03	47750	▲	1100	47900	46500	3524

< 그림 4 > 프로그램 초기화면

```
Public Sub Inet1_StateChanged(ByVal State As Integer)
    Dim strURL As String

    strURL="http://stock.naver.com/iop/stocks_jisu.wai?type=3&search_item=1&search_guide=1
    &query=UBIPia"

    Select Case State
    Case icError ' 11
        Dim vtData2 As String ' 데이터 변수.
        vtData2 = Inet1.ResponseCode & ":" & Inet1.ResponseInfo
        MsgBox (vtData2, vbCritical, 오류발생)
    Case 12 'icResponseReceived ' 12
        Dim vtData As String ' 데이터 변수.
        Dim strData As String: strData = ""
        Dim bDone As Boolean: bDone = False
        strData = Inet1.OpenURL(strURL, icString)
```

< 그림 5 > 추가정보 수집 프로그램 소스 코드

첫 번째 단계로, 위 < 그림 5 >에서 프로그램소스에서와 같이 먼저 주가정보를 가지고 오기 위해서 에이전트가 비주얼 베이직의 인터넷 관련 액티브엑스 컴포넌트(ActiveX Component)를 이용하여 인터넷상의 주가정보 웹 페이지를 실시간으로 수집한다[8]. 인터넷 컨트롤의 OpenURL 메소드는 매개변수로 주소 값을 받아서 웹 페이지를 하이퍼텍스트 형태로 반환하여 원하는 웹 페이지를 사용자의 컴퓨터로 가지고 올 수 있게 해준다.

두 번째 단계로는 수집된 웹 페이지를 파싱(Parsing)을 통하여 필요한 주가정보만 추출하여야 한다. 이 단계에서는 주가 정보를 제공하는 각 사이트마다 형태가 다르기 때문에 사용자가 필요한 정보가 소스코드의 어디에 있는지를 파악하여 그 외의 다른 코드부분을 삭제하는 방법으로 필요한 데이터를 추출할 수 있다.

```

<!-- start: 검색-->
<table border=0 cellpadding=0 cellspacing=2>
<tr><td colspan=4><formmethod=getaction=http://stock.naver.com/naverstock></td></tr>
<tr> <td>
        <select name=command>
        <option value=profile selected>현재가</option>
        <option value=minute>시간별</option>
        <option value=day>날짜별</option>
        </select> </td>
        <td><input type=text name=code size=18 value=한솔엠닷컴></td>
        <td><input type=image src=http://static.naver.com/common/images/search.gif
border=0></td>
        <td><small><a href=/portfolioonClick=
"javascript:window.open('/findcode.html','codefinder','width=295,height=365,resizable=1,status=0,scrollbars=0'); return false;">코드찾기</a></small></td>
    
```

< 그림 6 > 주가정보 웹 페이지의 소스코드

```

Public Function Parsing()
    '현재가 추출
    If InStr(strRow, "<td bgcolor=#cacbe7><small>현재가</small></td>") <>
Then
    Line Input #1, strRow
    intPos = InStr(strRow, "</b>") - 1 '뒷부분 자르는 위치지정
    strField = Left(strRow, intPos)
    '뒷부분 자르기 - 즉 앞에서부터 뒤부분자르는 위치까지 추출
    intPos2 = InStr(strRow, "<b>") + 2 '앞부분자르는 위치지정
    strField = Right(strField, (Len(strField) - intPos2)) '앞부분 자르기

    result(0 + cnt) = strField '배열에 결과값 넣기
    cnt = cnt + 1
End Function
    
```

< 그림 7 > 웹 페이지 파싱 프로그램 소스 코드

위의 < 그림 6 >의 Html소스를 위의 < 그림 7 >소스 프로그램을 통하여 현재가 부분을 추출할 때 그 앞부분과 뒷부분의 Html 코드를 지정하는 방식으로 현재가 정보를 웹 페이지에서 추출할 수 있다.

세 번째 단계로는 추출된 정보를 데이터베이스에 저장시키고, 알고리즘을 학습시킬 수 있도록 속성(Attribute)과 분류(Class)별로 학습예제(Training Example)를 작성한다. 학습예제를 작성하기 전에 먼저 속성(Attribute)과 부류( Class)를 정의해야 한다. 본 연구에서는 다음에 설명할 4개의 속성을 사용하였다.

첫 번째 속성은 과거주가형태(type)로써 일정 시간간격의 현재가 추세를 9가지의 형태로 나타냈다. 계산방법은 < 그림 8 >과 같이 시간순으로 정렬한 k개의 현재가 data의 평균을 구하고, 시작data와 마지막 data, 중간 data를 평균과 비교하여 그 데이터의 형태를 정의한다.

$$AP = \frac{\sum_{i=1}^k PP_i}{k}$$

AP(Average Price) = 평균가,  
 PP(Present Price) = 현재가  
 k = 시간 간격 내의 데이터 갯수

< 그림 8 > k 개의 현재가 평균

이를 바탕으로 현재가를 아래와 같이 9개의 타입으로 나타낼 수 있다.

PPI=시작값, AV=평균, PPk=마지막 값으로 정의하였을 때 다음과 같이 나타낼 수 있다.

- 상승형 : ( PP1 < AV < PPk ), 하락형 : ( PP1 > AV > PPk ), 보합형 : ( PP1 = AV = PPk )
- 상승하락형 : ( PP1, PPk < AV < PPk/2 ), 하락상승형 : ( PP1 & PPk > AV > PPk/2 )
- 상승보합형 : ( PP1 < AV < PPk/2 = PPk ), 하락보합형 : ( PP1 > AV > PPk/2 = PPk )
- 보합상승형 : ( PP1 = PPk/2 < AV < PPk ), 보합하락형 : ( PP1 = PPk/2 > AV > PPk )

두 번째 속성은 일정 시간간격에서 발생한 체결량으로 거래량의 변동을 나타내는 지수가 된다. 세 번째 속성은 매도/매수호가를 속성으로 사용하였는데, 이는 현재가와 비교하여 매도세력과 매수세력 중 어느 쪽이 우위인지를 나타내는 지수가 된다. 네 번째 속성은 전일대비 등락율로 이는 현재가의 비교자료가 된다. 위에서 설명한 속성 외에도 종합주가지수나 코스닥 지수, 나스닥지수나 관련업종의 지수 등을 속성으로 선택할 수도 있으나, 본 연구에서는 위의 네 가지 속성만을 사용하여 에이전트의 의사결정 트리를 생성하였다.

다음으로 의사결정 트리가 예측하고자 하는 부류를 정의하여야 한다. 이는 사용자가

예측하고자 하는 것이 현재부터 일정시간 후까지의 등락율의 평균이 될 수도 있고, 현재부터 일정시간 후까지의 등락율의 형태등 여러 가지가 있겠으나, 본 연구에서는 속성으로 사용된 단기주가추세형태의 미래 예측치를 부류로 사용하고자 한다. 이는 예측시점으로부터 일정 기간 후까지의 주가추세의 형태를 예측하고자 하는 것으로서 현재 추세가 하락형 또는 보합형일 때 미래예측추세가 상승추세로 반전하는 시기가 에이전트가 제시하는 매수시점이고, 현재추세가 상승형 또는 보합형이나 미래예측추세가 하락형일 때가 매도시점을 나타내는 것이다.

네 번째 단계로는 수집된 데이터를 기초로 의사결정 트리를 생성한다.

아래 < 표 1 >은 2000년 5월 28일부터 29일까지의 한솔 엠닷컴의 주가정보를 에이전트가 인터넷을 통해 주가정보사이트로부터 실시간으로 가져온 데이터를 DB에 저장한 것의 일부분이다.

< 표 1 > 주가정보 테이블

날짜	거래시간	현재가	전일비	시가	고가	저가	매도호가	매수호가	거래량
2000-05-28	14:04	6980	+680	6500	7000	6500	6980	6970	2199685
2000-05-28	14:05	6980	+680	6500	7000	6500	6980	6970	2203747
2000-05-28	14:06	6990	+690	6500	7000	6500	6990	6980	2217695
2000-05-28	14:06	6990	+690	6500	7000	6500	6990	6980	2217695
2000-05-28	14:07	6990	+690	6500	7000	6500	6990	6980	2219603
2000-05-28	14:08	6970	+670	6500	7000	6500	6980	6970	2222295
2000-05-28	14:34	6920	+620	6500	7000	6500	6920	6900	2353709
2000-05-28	14:37	6860	+560	6500	7000	6500	6900	6890	2377529
2000-05-28	14:37	6880	+580	6500	7000	6500	6880	6870	2380709
2000-05-28	14:39	6900	+600	6500	7000	6500	6900	6890	2381590
2000-05-28	14:39	6900	+600	6500	7000	6500	6900	6890	2381590
2000-05-28	14:43	6900	+600	6500	7000	6500	6900	6890	2448097
2000-05-28	15:00	6930	+630	6500	7000	6500	6930	6920	2579764
2000-05-29	09:01	6000	-460	6000	6000	6000	0	0	7474
2000-05-29	09:02	5950	-510	6000	6000	5950	5950	5940	24432
2000-05-29	09:03	5910	-550	6000	6000	5900	5910	5900	32374
2000-05-29	09:03	5910	-550	6000	6000	5900	5910	5900	32374
2000-05-29	09:03	5910	-550	6000	6000	5900	5910	5900	32374
2000-05-29	09:05	5910	-550	6000	6000	5890	6000	5910	43383

위의 < 표 1 >을 보면 10개의 컬럼이 있는데 여기서 현재가와 매도호가, 매수호가와 거래량 항목이 의사결정 트리를 생성하는데 사용된다.

이 데이터를 기초로 에이전트는 수집한 데이터를 학습할 수 있도록 자료를 가공하여 속성과 부류를 정의하고, 학습데이터로 변환시킨다.

여기서 중요한 것은 속성과 부류를 어떻게 정의하느냐에 따라서 학습된 의사결정 트리의 정확도에 영향을 미칠 수 있으므로 여러 속성을 정의하고, 학습단계를 거쳐서 종 목별로 정확도가 높은 속성을 선택할 수 있도록 한다. 본 연구에서는 위에서 정의한 4 개의 속성과 1개의 부류로 학습 데이터를 작성한다. 이 데이터는 의사결정 트리를 생성하는데 필요한 속성들과 부류를 정의하고 그런 속성과 부류 데이터를 일정한 형식

에 맞게 입력하는 역할을 한다

다음 < 표 2 >를 보면 5개의 속성이 있는데 ‘과거가타입’ 속성은 이산적 속성으로 9개의 이산 값의 집합인 것을 알 수 있다. 그리고 ‘현재가’와 ‘매도호가’, ‘매수호가’, ‘체결량’은 연속형 속성이고, 부류인 ‘미래예상가타입’은 ‘과거가타입’과 같은 이산값의 집합이다. @data 아래부터는 학습예제 데이터를 나타낸다. ‘상승형, 249500 , 249500 , 250000 , 41720 ,상승하락형’ 의 의미는 과거 일정 시간간격 간의 현재가 추세는 상승형이고, 현재가는 249500, 매도호가는 249500, 매수호가는 250000이며, 일정 시간간격사이의 거래 체결량은 41720일 때 현재부터 일정 시간간격 후까지의 현재가 추세는 상승하락형임을 의미한다.

< 표 2 > 학습예제 데이터

```
@relation training
@attribute 과거가타입 {상승형,하락형,보합형,상승하락형, 하락상승형, 상승보합형, 보합상승형, 하락보합형, 보합하락형}
@attribute 현재가 real
@attribute 매도호가 real
@attribute 매수호가 real
@attribute 체결량 real
@attribute 미래예상가타입 {상승형,하락형,보합형,상승하락형, 하락상승형, 상승보합형, 보합상승형, 하락보합형, 보합하락형}
@data
상승형, 249500 , 249500 , 250000 , 41720 ,상승하락형
상승형, 250000 , 249500 , 250000 , 38340 ,보합하락형
하락상승형, 249500 , 249500 , 250000 , 38340 ,상승하락형
하락상승형, 250000 , 249500 , 250000 , 38340 ,보합하락형
보합상승형, 249500 , 249500 , 250000 , 38310 ,상승하락형
하락상승형, 249500 , 249500 , 250000 , 38320 ,상승하락형
하락상승형, 249500 , 249500 , 250000 , 37930 ,상승하락형
하락상승형, 249500 , 249500 , 250000 , 38210 ,상승하락형
하락상승형, 249500 , 249500 , 250000 , 38700 ,상승하락형
하락상승형, 249500 , 249000 , 249500 , 38360 ,상승하락형
```

다음의 < 표 3 > 부터 < 표 6 >까지 이러한 5개의 속성과 1개의 부류를 가지고 있는 1893개의 학습 데이터를 이용하여 의사결정 트리를 구성한 결과이다.

< 표 3 > 학습예제 데이터 정보

```
=== Run information ===
Instances: 1893
Attributes: 6
    과거가타입
    현재가
    매도호가
    매수호가
    체결량
    미래예상가타입
```

위의 < 표 3 >에서 Instances 부분은 학습데이터의 수로서 현재 1893개의 학습예제 데이터가 있는 것을 알 수가 있다. 다음으로 Attributes 부분은 6개의 속성을 가지고

있는 것을 보여주는데, 여기서는 과거가 타입, 현재가, 매도호가, 매수호가, 체결량의 속성과 미래예상가타입의 부류를 나타내고 있다.

< 표 4 > 학습결과

```

C4.5 Algorithm
-----
매도호가 <= 250000
| 과거가타입 = 상승형
| | 매도호가 <= 249000
| | | 매도호가 <= 248500: 상승하락형 (11.0)
| | | 매도호가 > 248500
| | | | 체결량 <= 41050: 하락상승형 (5.0/1.0)
| | | | 체결량 > 41050
| | | | | 현재가 <= 249000: 상승보합형 (4.0/2.0)
| | | | | 현재가 > 249000: 상승하락형 (13.0/4.0)
| | 매도호가 > 249000
| | | 체결량 <= 41100: 하락상승형 (13.0/1.0)
| | | 체결량 > 41100
| | | | 체결량 <= 43800
| | | | | 매도호가 <= 249500
| | | | | | 체결량 <= 42190: 상승하락형 (8.0/3.0)
| | | | | | 체결량 > 42190: 하락상승형 (10.0/3.0)
| | | | | 매도호가 > 249500: 하락상승형 (7.0/1.0)
| | | | | | 체결량 > 43800: 상승하락형 (6.0)
    
```

위의 < 표 4 >는 학습예제를 가지고 의사결정 트리를 구성한 것으로서 제일 위에 있는 상위노드가 root이고, 그 아래쪽에 보이는 것들은 의사결정 트리의 노드로서 의사결정의 분기점이 되는 곳이다. 그리고, 제일 오른쪽 부분은 의사결정 트리의 leaf로서 뿌리(root)로부터 다른 노드로의 경로는 일련의 질문과 해답에 대응이 된다.

위의 첫 번째 가치를 해석하면 매도호가 250000보다 작거나 같을 때 과거가 타입이 상승형이고, 매도호가 249500 보다 작거나 같을 때 매도호가 248500보다 작거나 같으면 미래 예상가 타입이 상승하락형인 학습예제 데이터가 11개 있다는 의미이다.

< 표 5 > 의사결정트리 분석결과

```

=== Evaluation on training set ===
=== Summary ===

```

Correctly Classified Instances	1684	88.9593 %
Incorrectly Classified Instances	209	11.0407 %
Mean absolute error	0.038	
Root mean squared error	0.1379	
Relative absolute error	22.0152 %	
Root relative squared error	46.9363 %	
Total Number of Instances	1893	

위의 < 표 5 >를 보면 학습예제로부터 생성된 의사결정 트리의 분석결과를 보여준다.

여기에서는 1893개의 학습예제를 가지고 의사결정 트리를 구성하였고, 그중 88.9593 % 인 1684개의 예제가 생성된 의사결정 트리에 적합하였으며, 11.0407 % 인 209개의 예제가 부적합하였다.

다음 단계로는 위의 단계에서 생성된 의사결정 트리를 규칙 기반(Rule-Based)으로 만들어서 데이터베이스에 저장시킨다. 위의 의사결정 트리는 다음 <표 3.6>과 같은 주가 등락 규칙으로 변환할 수 있다.

< 표 6 > 주가 등락 규칙

<p>Rule 1 :</p> <p>If 매도호가&lt;=250000 And 과거가타입=상승형 And 매도호가&lt;=249000 And 매도호가&lt;=248500 then 상승하락형 (11.0)</p> <p>Rule 2 :</p> <p>If 매도호가&lt;=250000 And 과거가타입=상승형 And 매도호가&lt;=249000 And 매도호가&gt;248500 And 체결량&lt;=41050 then 하락상승형 (5.0/1.0)</p> <p>Rule 3 :</p> <p>If 매도호가&lt;=250000 And 과거가타입=상승형 And 매도호가&lt;=249000 And 매도호가&gt;248500 And 체결량&gt;41050 And 현재가&lt;=249000 then 상승보합형 (4.0/2.0)</p> <p>Rule 4 :</p> <p>If 매도호가&lt;=250000 And 과거가타입=상승형 And 매도호가&lt;=249000 And 매도호가&gt;248500 And 체결량&gt;41050 And 현재가&lt;249000 then 상승하락형 (13.0/4.0)</p>
---

위의 단계를 마친 에이전트는 제공되는 주가정보를 사용하여 규칙 데이터베이스에서 적합한 규칙을 낸 후 가까운 미래의 등락을 예측하여 사용자에게 적절한 매도/매수 시점을 통보할 수 있다.

## 4. 테스트 결과

### 4.1 구현 범위

여기서는 앞서 설명한 단기주가예측 에이전트 시스템의 구현을 위한 프로그램 작성 및 수행에 대해서 설명한다.

먼저 프로그램에 사용된 개발환경은 다음과 같다.

- ① 운영체제 : Windows 2000
- ② Database : Ms Access 2000
- ③ 개발 소프트웨어 : Ms Visual Basic 6.0
- ④ Java JDK 1.3.2
- ⑤ Data Mining Tool : Weka 3.1.8

시스템의 범위는 인터넷상에서 주가정보 웹 페이지를 가지고 오는 모듈과 그 웹 페이지에서 주가정보만을 추출해서 데이터베이스에 저장하는 모듈, 저장된 주가정보를 이용하여 학습예제를 생성하는 모듈, 생성된 학습예제를 가지고 의사결정 트리를 생성하는 모듈을 구현하였다.

## 4.2 테스트 내용 및 결과

주가를 예측하고자 할 때 중요한 사항은 주가에 영향을 주는 속성정보들을 많이 확보하는 것이 중요하고, 두 번째로는 예측하고자 하는 시간 간격이다. 기존 주가정보를 학습예제 데이터로 변환하는데 예측시간간격에 따라서 생성되는 의사결정 트리의 적합도와 테스트 데이터의 예측 정확도에 많은 영향을 준다. 이는 예측 시간 간격이 너무 짧으면 사소한 변동도 의사결정 트리의 생성에 영향을 줘서 복잡도를 증가시키기 때문이다. 그러므로 본 논문에서는 같은 데이터를 여러 예측시간간격으로 학습예제를 생성시켜 어느 정도의 예측시간 간격이 에이전트의 정확도를 높이는지에 관하여 테스트하고자 한다. 학습 데이터로는 위 < 표 1 >에서 제시한 한솔 엠닷컴의 주가정보를 에이전트를 이용하여 인터넷상에서 수집하여 가공한 데이터를 사용하였다. 생성된 의사결정 트리의 예측 정확도를 구하기 위하여 학습데이터의 마지막 10% 부분을 이용하여 테스트 데이터로 사용한다.

의사결정 트리의 속성으로는 위에서 설명한 과거주가추세형태와 현재가, 매도호가, 매수호가, 거래 체결량을 사용하였고 예측할 부류로는 미래주가추세형태를 사용하였다.

아래 < 그림 9 >과 < 그림 10 >는 예측시간간격을 10분부터 180분까지 증가시키면서 의사결정 트리를 생성하여 트리의 적합도와 테스트 데이터의 예측 정확도를 나타낸 것이다. 이 테스트 결과를 보면 예측시간간격이 증가하면서 트리의 적합도도 같이 증가하는 것을 볼 수 있다. 이는 시간간격이 너무 짧으면 의사결정 트리를 생성하는데 사소한 주가의 변동까지 포함하므로 복잡도가 증가하고, 전체적인 주가의 흐름이 왜곡되어 나타난다고 볼 수 있다. 그리고 예측 정확도는 120분까지는 낮은 예측 정확도를 보이며 일관성이 없지만 130분부터 180분까지는 평균 30%이상의 예측 정확도를 보이고 있다. 이 테스트의 결과를 보면 가장 높은 예측 정확도가 예측 시간간격을 150분으로 하였을 때의 41.2214%로 실제 주식거래의 참고자료로 사용하기에는 부족한 것을 알 수 있다. 이는 의사결정 트리를 생성하기 위해 사용한 속성들이 실제 주가에 끼치는 영향이 그리 크지 않다는 것을 보여주는 것이다. 그러나 다른 정보, 예를 들면 매도/매수 잔량, 동종업종의 등락율, 미국 나스닥의 동종업종 지수 등을 의사결정 트리의 속성으로 추가하였을 때에는 더욱 높은 예측 정확도를 기대할 수 있다.

또한 생성된 규칙별로 예측 정확도를 계산하여 그중 높은 예측 정확도를 갖는 규칙에 부합하는 정보가 들어왔을 경우 에이전트가 등락을 예측하여 사용자에게 매도/매수 시점을 통보하게 할 수도 있다.



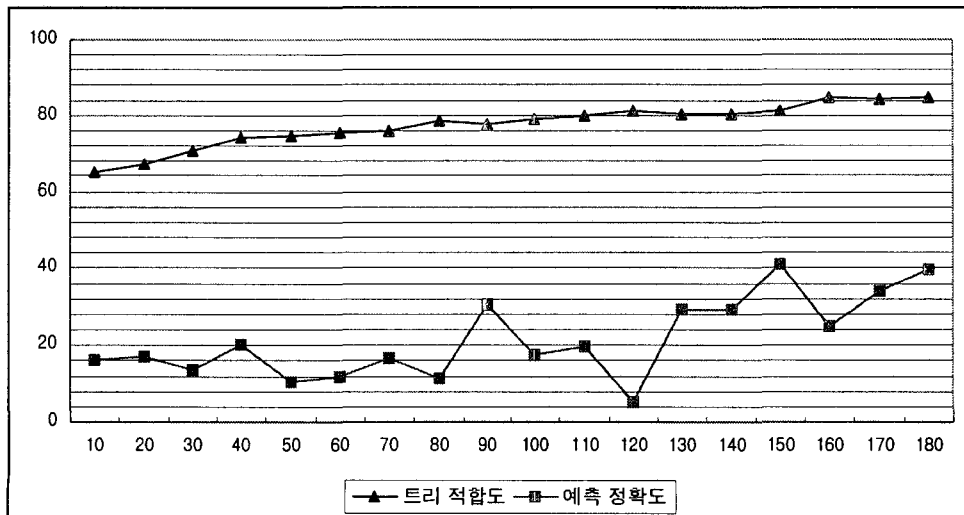
**SPF Agent System**

테스트 결과보기  
 초기화면으로

학습에이전트 테스트 날짜 | 2003-(05월 15일)  
 총 테스트데이터 수 | 38,916  
 총 테스트 수 | 4,716

시간간격	예제 수	테스트 수	트리 적합도	예측 정확도
10	2332	262	65.0943	16.0305
20	2312	262	67.128	17.1756
30	2292	262	70.7679	13.3588
40	2272	262	74.2077	20.229
50	2252	262	74.3783	10.687
60	2232	262	75.5272	11.8321
70	2212	262	76.1302	16.4122
80	2192	262	78.6496	11.4504
90	2172	262	77.9006	30.5344
100	2152	262	79.2286	17.5573
110	2132	262	79.8311	19.8473
120	2112	262	81.0506	5.3435
130	2092	262	80.2103	29.3893
140	2072	262	80.3571	29.3893
150	2052	262	81.5302	41.2214
160	2032	262	84.5472	24.8092
170	2012	262	84.1451	33.9695
180	1992	262	84.5382	39.6947

< 그림 9 > 테스트 결과



< 그림 10 > 예측 시간 간격별 예측 정확도

## 5. 결론 및 추후 연구과제

본 연구에서 제시한 학습 에이전트를 이용한 주가예측은 개별 종목별로 학습을 시켜서 종목별로 정확한 예측을 가능하게 해주고, 실시간으로 매도/매수 시점에 대한 피드백(feedback)이 가능하므로 연속된 학습과 예측, 오차의 수정을 보다 쉽게 할 수 있다는 장점을 가지고 있다. 결과적으로 인간의 분석 능력에는 한계가 있으므로 에이전트를 이용하면 인간을 대신하여 여러 종목의 주가의 추세를 지속적으로 관찰하고 판단하게 함으로써 주가에 영향을 주는 변동을 실시간으로 탐지할 수 있다. 다시 말하면 많은 종목들의 변동을 동시에 실시간으로 확인할 수 있고, 다른 종목간의 연관성 및 상관관계도 학습 에이전트를 이용하면 쉽게 발견할 수 있다.

본 연구에서 제시한 SPFA(Stock Price Forecasting Agent)는 데이터의 수집, 분류, 저장, 학습, 예측, 재학습까지 자동으로 해줄 수 있는 개인화 된 에이전트이다.

사실 인터넷상의 정보들은 비 정형화된 데이터, 즉, 각 사이트들마다 다른 형태의 웹 페이지를 제공하므로 에이전트의 데이터 수집에는 한계가 있다. 그러나, 이런 정보를 수집하는 에이전트에 각 사이트의 형태에 맞는 필터를 제공하면 이러한 문제점은 해결될 수 있을 것이다. 에이전트를 이용할 때의 또 다른 중요한 문제는 학습부분으로 본 연구에서는 의사결정 트리 알고리즘을 사용하였지만 해당 도메인에 맞는 여러 학습 알고리즘을 사용하여 에이전트학습의 정확도를 높일 수 있었다.

또한, 의사결정 트리 생성시 사용하는 속성들을 미리 정의하여 사용하였으나, 여러 속성 중 에이전트가 유전자 알고리즘이나, 인공 신경망 등의 알고리즘을 이용하여 어떤 속성들을 사용하여야 의사결정 트리의 정확도를 높일 수 있는지 속성에 대한 선 처리(Pre-Process)를 한 후에 의사결정 트리를 생성한다면 더욱 예측 정확도를 높일 수 있을 것이다. 마지막으로 본 연구에서는 주가예측 분야에 학습능력을 가진 에이전트를 적용하였지만, 추후 여러 분야로 응용할 수 있는 에이전트를 개발하고, 그런 학습된 에이전트들을 서비스하는 에이전트 서버를 운용하여 사용자의 요구를 적합한 에이전트를 수행시킬 수 있는 에이전트 포털 사이트에 관한 연구로 확장할 수 있을 것으로 예측된다.

## 6. 참고 문헌

- [1] 백준걸, 김강호, 김창욱, 김성식, “실시간 기계 상태 데이터베이스에서 데이터 마이닝을 위한 적응형 의사결정 트리 알고리즘”, 대한산업공학회지, 2000.
- [2] 임도형, 이일병, “신경망을 사용한 매도/매수 주식 종목 선정”, 대한산업공학회지, 2000.
- [3] 오성민, 김성집, “주식의 상산가시 잔량과 일일거래량의 관계를 통한 주가의 단기 예측에 관하여(kospi 200종목을 중심으로)”, 대한산업공학회지, 2000.
- [4] Thomas Dean, 김진형 공역, “인공지능 이론 및 실제”, 사이텍미디어, 1998.
- [5] Gerhard Weiss, “Multi Agent Systems : A Modern Approach to Distributed Artificial Intelligence”, MIT Press, 1999.

- [6] Ianh Witten, Eibe Frank, "Data Mining", Morgan Kaufmann, 2000.
- [7] Alper Caglayan, Colin Harrison, "AGENT", Wiley&Sons, 1997.
- [8] Dan Appleman, "Developing COM/ActiveX components with Visual Basic 6", SAMS, 1999.
- [9] Adriaans, P. and Zantinge, D, "Data Mining", Addison-Wesley, 1996.
- [10] Quinlan, J. R., "C4.5: Programs for Machine Learning", Morgan Kaufmann Publ
- [11] Adaptive Recommendation System, Jongwoo Lee & Byoung-Tak Zhang, 2nd International Symposium on Advanced Intelligent Systems, Korea, pp. 321-325, 2001.
- [12] Genetic Mining of HTML Structures for Effective Web-Document Retrival, Kim, S. and Zhang, B.-T., Applied Intelligence, 2001.
- [13] Evolutionary Learning of Web-Document Structure for Information Retrieval, Sun Kim & Byoung-Tak Zhang, The 2001 Congress on Evolutionary Computation (CEC2001), pp. 1253-1360, 2001.

### 저 자 소 개

서 장 훈 : 명지대학교 산업공학과를 졸업, 동 대학원 산업공학과 석사를 취득 현재는 산업공학 박사과정이다. 주요관심 분야는 e-Business 분석, SCM, 품질공학, Data-Mining. 6sigma

장 현 수 : 명지대학교 산업공학과 석박사 취득, 현재 경기공업대 산업경영시스템과 교수로 재직중이다. 주요관심 분야는 품질공학, 동시공학, e-Biz 경영, Internet 경영.