The Korean Communications
in Statistics Vol. 10 No. 3, 2003
pp. 845-857

A Decision Tree Algorithm using Genetic Programming

Chongsun Park[1], Young Kyong Ko[2]

Abstract

We explore the use of genetic programming to evolve decision trees directly for classification problems with both discrete and continuous predictors. We demonstrate that the derived hypotheses of standard algorithms can substantially deviated from the optimum. This deviation is partly due to their top-down style procedures. The performance of the system is measured on a set of real and simulated data sets and compared with the performance of well-known algorithms like CHAID, CART, C5.0, and QUEST. Proposed algorithm seems to be effective in handling problems caused by top-down style procedures of existing algorithms.

Keywords : Classification, Decision Tree, Genetic Programming, Mixed Predictors.

1. 서론

기계학습(Machine Learning)방법 중의 하나인 의사결정나무(Decision Tree)는 의사결정규칙(Decision Rule)을 도표화하여 관심대상을 몇 개의 소집단으로 분류하거나 예측하는 분석방법으로서, 결과의 해석이 쉽고 새로운 자료에 대한 모형 적합이 용이하여 데이터마이닝(Data Mining) 기법으로 널리 사용되고 있다. 의사결정나무 알고리즘에 대한 단점 중의 하나는 그리디(greedy) 알고리즘을 이용하여 분리가지를 선택한다는 것이다. 이 방법은 분리가 이루어지는 마디에서의 변수 선택이 이후의 분리에 대한 고려 없이 결정되는 일련의 하향(top-down)방식이기 때문에 이전 마디에서의 분리결과가 다음 분리 마디에 영향을 미치게 된다. 이런 측면에서 기존 알고리즘은 최적 의사결정나무의 형성을 보장하지 못하고 분류의 예측력에 나쁜 영향을 미칠 수 있다.

전역 최적해를 찾지 못하는 하향식 알고리즘의 단점을 보완하는 방법으로는 Bagging(Breiman, 1997)이나 Boosting과 같이 표본 재추출을 이용하여 다수의 모형을 결합하는 방법과 유전알고리즘을 이용한 방법이 있다. 유전알고리즘(Genetic Algorithm)은 유전학과 자연진화를 흉내 낸 적응 탐색법으로, 기존의 결정론적 알고리즘으로 해를 찾기 어려운 큰 탐색 공간을 가진 문제에 적용된다. 의사결정나무의 형성방법으로 유전알고리즘을 응용한 유전프로그래밍(Genetic Programming)을 이용한 방법이 Koza(1991)에 의하여 제시되었다. 유전프로그래밍을 이용하여 의사결정나무를 형성하는 연구들은 주로 기존의 하향 방식의 알고리즘을 벗어난 최적의 의사결정나무의 형성을 시도한 것으로 계산상으로 복잡하다는 유전프로그래밍의 단점을 보완하는 것과 검색 공간을 넓혀

1) Associate Professor, Department of Statistics, Sungkyunkwan University, Seoul, 110-745, KOREA
E-mail : cspark@skku.edu
2) Graduate Student, Department of Statistics, Sungkyunkwan University, Seoul, 110-745, KOREA
E-mail : youngko@hotmail.com

정확하면서도 간단한 의사결정나무를 형성하는 방법에 관련된 연구들(Papagelis 등, 2000; Bot 등, 2000; Nikolaev 등, 1997; Fu, 2001)이 진행되어 왔다.

본 논문에서는 기존의 탑-다운 방식을 벗어나 유전프로그래밍을 이용한 의사결정나무 형성방법으로 Papagelis 등(2000)에 의하여 제시된 알고리즘을 개선하여 이산형인 설명변수뿐만 아니라 연속형인 설명변수들도 포함된 자료에 적용이 가능한 방법을 고려하였다. 각 마디에서의 분리는 C4.5와 같이 이산형인 경우는 수준의 수만큼, 연속형인 경우는 이분하는 방법을 택하였다. 더불어 간단하면서도 정확한 의사결정나무 형성을 위한 적합도 함수를 알아보고 기존의 알고리즘들과 비교하였다.

본 논문의 제 2 장에서는 의사결정나무에 관한 내용을 설명하였고, 제 3 장에서는 유전프로그래밍을 이용하여 이산형 및 연속형 설명변수를 모두 포함하는 의사결정나무를 형성하는 알고리즘을 소개하였다. 제 4 장에서는 유전프로그래밍을 이용한 의사결정나무 형성 알고리즘과 실제 실행 결과가 제시된다. 그리고 마지막으로 제 5장에서는 본 논문을 정리하였다.

2. 의사결정나무

의사결정나무는 선형성이나 정규성 또는 등분산성 등의 가정을 필요로 하지 않는 비모수적인 방법이다. 분류나 예측을 위한 목표변수를 Y라 하고 목표변수와 관련된 p개의 설명변수를 $X_1, X_2, \ldots, X_p$라 할 때 의사결정나무는 목표변수의 성격에 따라 분류나무(Classification Tree)와 회귀나무(Regression Tree)로 나눌 수 있다. 분류나무는 목표변수 Y가 이산형인 경우에 회귀나무는 목표변수 Y가 연속형 변수인 경우에 사용된다. 본 연구에서는 목표변수가 이산형이고 설명변수에는 제약이 없는, 즉 이산형과 연속형이 모두 포함된 분류나무에 대하여 고려하기로 한다. 의사결정나무를 만드는 데 사용되는 알고리즘은 거리를 측정하는 방법이나 정지규칙, 가지치기방법에 따라 여러 가지가 제기되었는데 CHAID (Chi-squared Automatic Interaction Detection: Kass, 1980), CART(Classification And Regression Trees: Breiman 등, 1984), QUEST(Loh 등, 1997), C4.5 (Quinlan, 1986, 1993) 등이 있다.

2.1 의사결정나무 형성과정

의사결정나무를 이용한 분석은 일반적으로 의사결정나무의 형성, 가지치기, 타당성 평가, 해석 및 예측의 단계로 이루어진다.

- 의사결정나무 형성단계 : 분석의 목적과 자료구조에 따라 적절한 분리기준과 정지규칙을 지정하여 의사결정나무를 얻는다.
- 가지치기 단계 : 분류오류(classification error)를 크게 할 위험이 있거나 부적절한 추론 규칙(induction rule)을 가지고 있는 가지를 제거한다.
- 타당성 평가단계 : 이익도표(gains chart)나 위험도표(risk chart) 또는 검정용 자료(testing data)에 의한 교차타당성(cross validation) 평가 등을 이용하여 의사결정나무를 평가한다.
- 해석 및 예측단계 : 의사결정나무를 해석하고 예측모형을 설정한다.

일반적인 의사결정나무 형성과 정지, 가지치기의 개념과 기준은 다음과 같다.

1) 분리기준(Splitting Criterion)

분리기준은 하나의 부모마디로부터 자식마디들이 형성될 때 변수의 선택과 범주의 병합이 이루어지는 기준을 의미한다. 즉 어떤 변수를 이용하여 어떻게 분리하는 것이 목표변수의 분포를 가장 잘 구별해 주는지를 파악하여 분리가 발생한다. 목표변수의 분포를 구별하는 정도를 순수도나 기타 다른 기준으로 측정하는데, 분리기준으로 마디의 순수정도를 나타내는 지니계수와 동질성을 나타내는 엔트로피지수 등이 주로 사용된다.

2) 정지규칙(Stopping Rule)

경계 없이 확장되는 나무구조는 시간을 많이 필요로 하고, 나무구조를 이해하기 힘들게 하며 자료를 과적합(overfitting)시킨다. 나무구조의 크기는 정지규칙이나 성장한계에 의해서 조절될 수 있다. 일반적인 정지규칙은 성장할 수 있는 최대깊이(maximum depth)를 제한하는 것이다.

3) 가지치기(Pruning)

지나치게 많은 마디를 가지는 의사결정나무는 새로운 자료에 적용될 때 앞서 언급한 과적합 문제를 일으켜 예측오차가 커질 가능성이 있기 때문에 형성된 의사결정나무에서 적절하지 않는 마디를 제거하여 적당한 크기의 구조를 가지는 의사결정나무를 최종적인 예측모형으로 선택하는 것이 바람직하다. 일반적으로 나무구조를 최대크기까지 성장시킨 뒤 어떤 기준에 의하여 정확도를 떨어뜨리지 않는 가장 작은 크기로 가지를 치게 된다.

3. 유전프로그래밍을 이용한 의사결정나무 형성

3.1 유전프로그래밍

유전프로그래밍(Genetic Programming: GP)은 유전알고리즘에서 파생된 형태로서, 프로그램 공간에서 유전 학습모델을 연장하여 컴퓨터 프로그램 문제를 해결할 수 있도록 Koza(1991)에 의해 개발되었다. 유전프로그래밍에서의 염색체는 다수의 함수(function)와 터미널(terminal)로 구성되는 계층적 나무 구조(hierarchical tree structure)를 가지는 컴퓨터 프로그램을 뜻하고 그 길이가 고정되어있지 않다. 유전프로그래밍은 나무 모양의 개체구조를 사용하며, 함수마디(function node)와 터미널 마디(terminal node)를 이용하여 원하는 함수를 표현한다. 각 함수마디는 산술 연산, 논리연산 등을 의미하고, 터미널마디는 그 연산의 입력 값들을 의미한다. 유전프로그래밍은 로봇의 프로그램 생성, 게임의 프로그램, 패턴인식, 인공지능에 관한 다양한 문제해결 및 학습 등에 응용되고 있다.

3.1.1 유전프로그래밍의 기본 과정

유전프로그래밍의 기본적인 연산구조는 근본적으로 유전알고리즘의 연산과 비슷하다. 먼저 초기

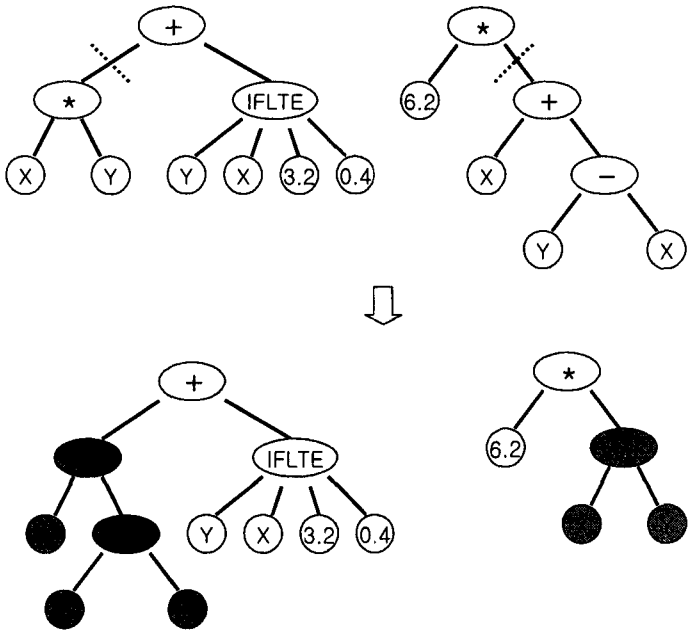
모집단이 임의로 생성되는데, 모집단 내의 염색체는 문제의 함수와 터미널로 이루어진 컴퓨터 프로그램이 된다. 염색체를 복호화하여 프로그램을 실행하고 문제 해결 능력에 따라 적합도 값을 할당한다. 초기 모집단이 평가된 후 재생산에 의해 새로운 집단이 만들어지고 다시 교배와 돌연변이를 통해 집단을 개선해 나가는 과정을 반복한다. 많은 세대가 지나는 동안 최적의 염색체(프로그램)가 집단에 나타나게 되고 이와 유사한 염색체들이 집단을 지배하게 된다.

1) 컴퓨터 프로그램인 초기 모집단 형성

유전프로그래밍의 초기모집단은 임의로 생성된 각각의 컴퓨터 프로그램들이다. 프로그램 나무의 중간마디에 나타나는 함수 집합(function set)은 일반적인 수학 함수나 조건부 연산자를 포함한다. 끝마디에 나타나는 터미널 집합(terminal set)은 변수와 상수들을 포함한다. 임의로 형성된 프로그램은 각각 다른 크기와 형태를 가진다. $F=\{f_1, f_2, \cdot\cdot\cdot, f_{N_{func}}\}$를 함수의 집합이라 하고, $T=\{a_1, a_2, \cdot\cdot\cdot, a_{N_{term}}\}$을 터미널의 집합이라고 하자. 여기서 N_{func}와 N_{term}은 각각 함수와 터미널 집합의 수를 나타낸다. 그리고 함수 집합 F의 각 함수 f_i는 $z(f_i)$를 인수로 갖는다. 그리고 집합 $C=F\bigcup T$ 라고 하자. 프로그램 나무의 뿌리에서 임의로 집합 F의 한 함수 f_i를 선택한다. 그 지점에서 $z(f_i)$만큼 분리가 형성된다. 형성된 각 분리에는 집합 C의 요소가 임의로 선택되는데, 함수가 선택되면 위 과정을 반복하고 터미널이 선택되면 그 점에서 끝낸다. 이 과정을 좌에서 우로 반복하여 하나의 초기 모집단이 형성된다. 이 과정을 필요한 초기모집단 개수만큼 반복한다.

2) 선택(Selection, Reproduction)

유전프로그래밍에서도 선택 연산자는 자연선택과 적자생존을 나타내는 중요한 연산자이다. 하나의 나무구조를 적합도에 근거하여 모집단으로부터 정해진 선택 방법에 따라 확률적으로 선택한 후, 선택한 개별 나무구조로 새로운 모집단을 형성한다.



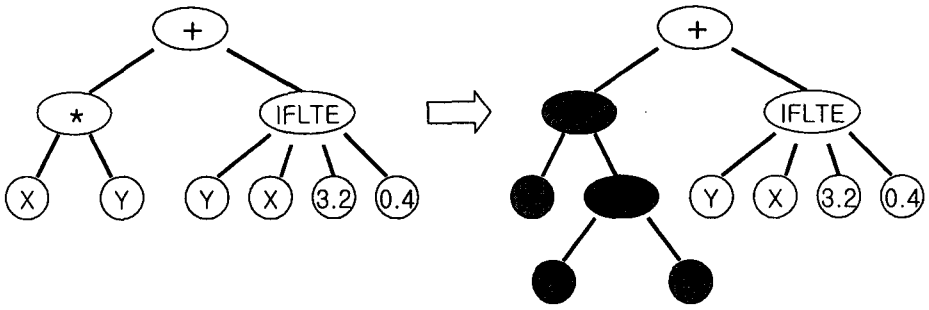
<그림 3-1> 유전프로그래밍에서의 교배

3) 교배(Crossover)

유전프로그래밍에서 교배 연산자는 2개의 부모염색체의 일부분으로 구성된 새로운 개체를 만든다. 유전알고리즘의 교배 연산자와의 차이점은 부모의 교배점을 각각 임의로 선택한다는 점이다. 교배는 유전프로그래밍에서 주된 연산자이고, 일반적으로 높은 확률로 수행된다. <그림3-1>은 유전프로그래밍에서의 교배연산자의 예이다.

4) 돌연변이(Mutation)

돌연변이 연산자는 모집단에 임의의 변화를 주게 된다. 유전프로그래밍에서 돌연변이 연산자는 임의의 마디를 선택한 후 그 마디를 제거하고 새로 생성된 부나무(subtree)구조를 삽입한다. <그림3-2>는 유전프로그래밍에서의 돌연변이의 예이다.



<그림 3-2> 유전프로그래밍에서의 돌연변이

그 이외에 역위, 편집, 캡슐화, 소거 등의 기타 연산자를 가질 수 있다.

3.2 기존 알고리즘들

유전알고리즘이 최적화 문제에 광범위하게 사용되어 왔지만 의사결정나무를 진화시키는 도구로 유전알고리즘 또는 유전프로그래밍을 제안한 연구는 많지 않았다. Koza(1991)는 Quinlan의 ID3(Quinlan, 1979) 알고리즘을 응용하여 유전프로그래밍이 분류나 패턴 인식을 위한 의사결정나무 형성에 사용될 수 있다는 것을 보였다. Koza는 의사결정나무 형성을 위해 나무구조의 염색체(chromosome)를 사용하였으며 터미널 집합으로 목표변수의 범주의 집합을 사용하였고, 함수 집합으로는 각 설명변수의 테스트 함수를 사용하였다. 적합도 함수로는 정확하게 분류된 학습용 자료의 수를 사용하였다.

Nikolaev와 Slavov(1997)는 전역적인 적합도 전망 구조(global fitness landscape structure)에 대하여 연구하였으며, 이것을 의사결정나무 형성에 적용하였다. 그들은 적합도 거리상관(fitness distance correlation)을 사용하였는데, 주어진 점에서 알려진 최적해에 쉽게 접근할 수 있게 해주는 것으로 알려져 있다. Papagelis 등(2001) 등은 유전프로그래밍을 이용한 의사결정나무 형성을 구체적으로 시도하였는데. 터미널 집합과 함수 집합의 구성은 Koza의 방법을 사용하였으며, 설명변수가 연속형인 경우 이산화 하여 사용하였다. 이 방법의 적합도 함수로는 다음이 사용되었다.

$$payoff(tree\ i) = CorrectClassified_i^2 \times \frac{x}{size_i^2 + x}$$

여기서 $tree\ i$는 모집단의 i번째 나무를 말하며, $CorrectClassified_i^2$은 i번째 나무의 정확하게 분류된 학습용 자료의 수를 의미한다. 위의 적합도 함수는 크기와 정확성을 동시에 고려한 것이다. 여기서 x는 임의의 큰 수이다. 그 외에도 Bot와 Longdon(2000)은 유전프로그래밍을 검정식에 여러 변수를 포함하는 선형 의사결정나무를 진화시키는데 사용하였으며, Fu(2001)는 초기 모집단을 C4.5와 같은 기존의 알고리즘을 이용하여 형성한 후 유전 연산자를 사용하여 의사결정나무를 진화시키는 방법을 제안하였다.

3.2.1 유전프로그래밍의 예측력을 높이기 위한 방법들

유전프로그래밍을 이용함으로써 발생할 수 있는 과적합 문제를 해결하고 예측력을 높이기 위해 다음과 같은 방법들이 제안되었다.

1) LIMITED ERROR FITNESS의 응용

Limited Error Fitness(LEF: Gathercole, 1998)는 오차 한계(error limit)를 도입하여 유전프로그래밍의 일반적인 지도 학습(supervised learning) 과정을 수정한 것이다. Bot(2000)등은 변형된 LEF를 사용하여 적합도 평가 과정에서 효율성을 증대시켰다. 유전프로그래밍을 이용한 의사결정나무 형성 과정에서 각 개체(나무구조)는 훈련용 자료의 개수만큼 평가가 된다. 즉 각 개체의 예상되는 분류(유전프로그래밍을 이용해서 형성한 의사결정나무를 통한 분류)는 각 훈련용 자료의

실제 분류와 비교된다. LEF는 현재까지 훈련용 자료의 오분류 수가 오차 한계보다 크다면, 남아 있는 자료를 모두 오차로 간주한다. 에러의 총수를 적합도로 사용하는 경우 이 방법은 열등한 개체는 모든 자료에 대해서 평가되지 않기 때문에 CPU 시간을 줄일 수 있다.

2) BLOAT문제의 해결

유전프로그래밍에서 각 개체는 시간이 지남에 따라 커지는 경향이 있으며 Koza(1992)는 이런 현상을 BLOAT라고 하였는데, BLOAT가 발생하면 각 개체를 평가하는 실행시간이 증가할 뿐만 아니라 나무구조를 이해하기도 힘들어지게 된다. 다음과 같이 깊이(depth)나 노드의 수(size)를 고려한 적합도 함수를 사용하여 BLOAT를 줄일 수 있다.

$$\text{Fitness} = \text{Fitness} + \text{depth(혹은 size)} \times \text{penaltyValue}$$

여기서 Fitness는 오분류된 학습용 자료의 수를 나타내고, depth와 size는 의사결정나무의 깊이와 크기를 나타낸다. PenaltyValue는 일종의 벌점으로 문제와 상황에 따라 적당한 값을 줄 수 있다.

3.3 제안된 알고리즘

본 절에서는 목표변수가 범주형이고 이산형 및 연속형 설명변수를 모두 포함하는 분류문제에 대하여 유전프로그래밍을 이용한 의사결정나무를 형성하는 새로운 방법을 제안하였다. 유전프로그래밍 관련 부분은 A.Qureshi의 GP system(GPSys)를 사용하였으며 이를 이용한 의사결정나무 형성과 관련해서는 Bot의 프로그램을 참조하였다. GPSys는 자바(Java) 언어로 되어 있으며, 토너먼트 선택을 이용한 엘리티즘(Elitism)을 사용한다.

<표 3-1>에 새로운 알고리즘에 사용된 기본적인 환경을 정리하였다. 터미널 집합과 함수집합은 Koza의 것을 그대로 사용하였으며 이산형 설명변수인 경우 범주의 수만큼 분리하였고 연속형인 경우에는 다음과 같이 변환하여 이분하였다.

$$\text{변환된 자료값} = \frac{(\text{자료값} - \text{해당변수의 최소값})}{(\text{해당변수의 최대값} - \text{해당변수의 최소값})}$$

<표 3-1> 유전프로그래밍을 이용한 의사결정나무 모수 설정

목적	학습용 자료를 정확하게 분류하는 것
터미널 집합 (Terminal Set)	목표변수의 분류값
함수 집합 (Function Set)	각 설명변수
선택 (Selection)	토너먼트 선택(토너먼트의 크기=7)
유전프로그래밍 파라미터	모집단 크기 = 300 반복 회수 = 30 세대변이(generation)수 = 100 엘리티즘(Elitism) 사용 교배(crossover)확률 = 0.9 돌연변이(mutation)확률 = 0.1
초기 모집단 형성	RAMPED_HALF_AND_HALF (RHAH)
종료 조건	BLOAT penalty를 포함한 fitness 함수

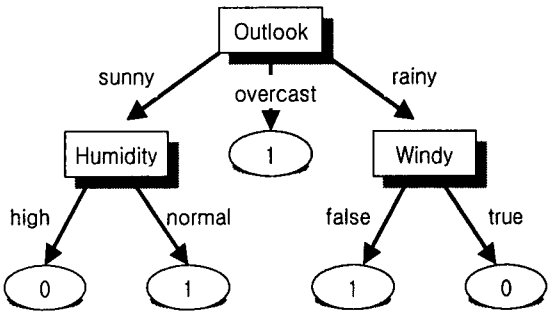
연속형 변수인 경우의 분리 기준값은 변환된 자료값의 최소값과 최대값 사이의 임의의 실수로 지정된다.

선택은 토너먼트 방식을 사용하였다. 토너먼트의 크기는 Bot(2000)가 제시한 7을 사용하였다. 교배와 돌연변이 확률은 <표 3-1>과 같이 설정하였고, 초기 모집단은 초기모집단 형성 시 50%는 모든 잎마디에서 지정된 깊이의 나무구조가 형성되고, 나머지 50%는 지정된 깊이 내에서 임의로 나무구조가 형성되는 RHAH방법을 이용하여 생성하였다. 종료조건은 유전프로그래밍의 효율을 높이기 위해 LIMITED ERROR FITNESS를 사용하였으며, BLOAT를 줄이기 위해 크기(size)에 penaltyValue를 주어 제한을 두었다.

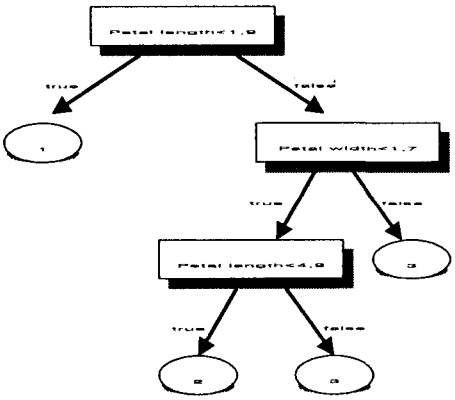
제시한 방법의 적합성 및 기존의 알고리즘들과의 비교를 위하여 전형적인 자료에 대한 결과를 간략하게 살펴보고 제 4 절에서 가상 및 실제자료들에 대하여 유전프로그래밍을 이용한 의사결정나무와 기존의 C5.0과 CART, CHAID 등에 의하여 형성된 나무들에 대한 교차타당성(Cross-Validation) 평가를 통한 오분류율 및 형성된 나무의 크기, 깊이 등을 비교하였다. 물론 유전프로그래밍을 이용한 의사결정나무가 계산상으로 복잡하여 다른 알고리즘에 비하여 상대적으로 시간이 많이 소요되는 단점이 존재한다.

3.3.1 모든 설명변수가 이산형인 경우(날씨자료에 대한 적용)

날씨자료(Quinlan, 1986)는 네 개의 설명변수(Temperature, Humidity, Outlook, Windy)와 두 가지 범주(1: positive, 0: negative)를 가지는 목표변수가 있으며, 설명변수 중 Temperature와 Humidity가 이산형인 자료와 연속형인 2개의 자료가 있다. 모든 설명변수가 이산형인 자료에 대하여 <그림 3-3>은 초기모집단수를 200으로 한 경우 두 번의 세대교체 후 생성된 의사결정나무 구조이다. 이 나무구조는 모든 학습용 자료를 정확하게 분류할 수 있고, C4.5 알고리즘을 이용한 결과와 동일하다. GPTree는 본 논문에서 제시된 유전프로그래밍을 이용한 의사결정나무 형성 방법을 의미한다.



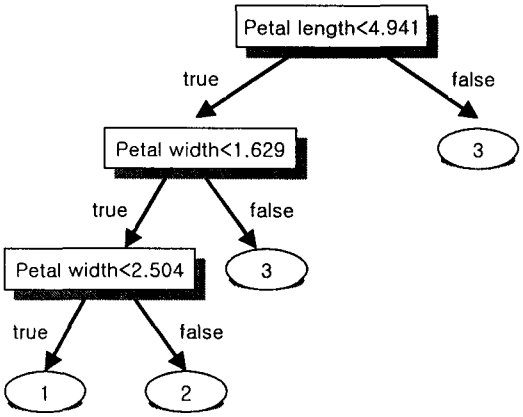
<그림 3-3> GPTree를 이용한 날씨자료 결과[1]
(모든 설명변수가 이산형인 경우)



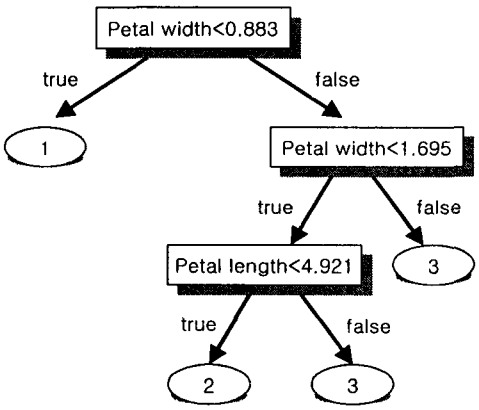
<그림 3-4> C5.0을 이용한 Iris 자료 결과

3.3.2 모든 설명변수가 연속형인 경우(Iris 자료에 대한 적용)

Iris 자료는 통계학과 관련된 문헌들에서 널리 알려진 자료이다. Iris의 3가지 종류에 대해 각각 50개의 자료가 있다. 설명변수의 수는 4개(sepal length, sepal width, petal length, petal width)이고, Iris의 종류를 나타내는 목표변수의 클래스(class)는 3개(Iris Setosa, Iris Versicolour, Iris Virginica)이다. 자료의 총수는 150개이다.



<그림 3-5> GPTree를 이용한 Iris 자료 결과[1]

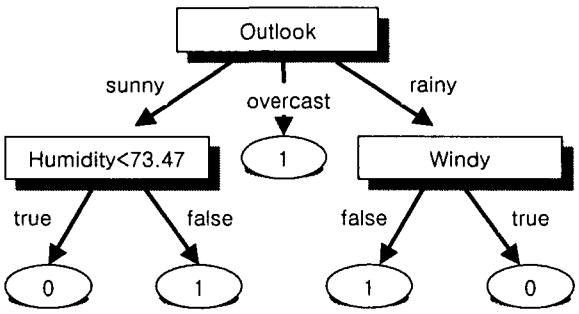


<그림 3-6> GPTree를 이용한 Ir 자료 결과[2]

Iris 자료에 대한 C5.0, CART, QUEST에서의 의사결정나무는 각각 상이하였다. 이 중 같은 정분류율(0.98)에서 가장 크기(node의 수)가 작은 것이 C5.0을 이용한 것이었는데, <그림 3-4>과 같다. 20번의 시행결과 가장 우수하게 나온 유전프로그래밍을 이용한 의사결정나무는 <그림 3-5>와 <그림 3-6>와 같다. 세 가지 의사결정나무 그림은 같은 정분류율(0.98)과 같은 크기를 가진다. <그림 3-6>은 C5.0, CART, QUEST의 뿌리마디(root node)에서 공통적으로 나온 petal length 변수를 사용하지 않고, petal width를 뿌리마디로 사용하였다. 정분류율에서는 같은 결과가 나왔지만, 유전프로그래밍은 기존의 탑-다운 방식으로는 생각할 수 없는 영역에 대한 검색도 실시해준다는 것을 확인할 수 있다.

3.3.3 설명변수가 이산형 혹은 연속형인 경우(날씨자료에 대한 적용)

다음으로 두 개의 설명변수(Outlook과 Windy)는 이산형이고, 나머지 두 개의 설명변수(Temperature와 Humidity)가 연속형인 날씨자료를 고려하였다. <그림 3-7>는 초기모집단수를 200으로 한 경우 다섯 번의 세대교체 후 생성된 의사결정나무 구조이다. 이 나무구조는 모든 학습용 자료를 정확하게 분류할 수 있고, 연속형 변수인 Humidity에서 분리 기준값이 차이가 나는 점을 제외하고는 C5.0 알고리즘을 이용한 결과와 동일하다. C5.0에서의 Humidity에서 분리 기준값은 75이다.



<그림 3-7> GPTree를 이용한 날씨자료 결과[2]
(설명변수가 이산형이거나 연속형인 경우)

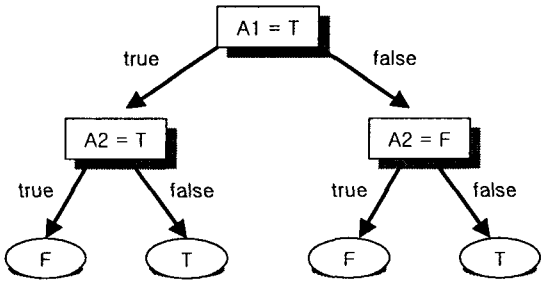
4. 가상 및 실제자료에 대한 적용 결과

4.1 가상자료

고려한 가상자료는 Papagelis 등(2001)에 인용되었으며 최적해가 알려져 있으나 기존의 그리디 알고리즘들이 대부분 탐색에 실패하는 것으로 알려진 것이다. 목표변수의 값은 A1 변수와 A2 변수의 *XOR*(exclusive or)함수에 의해 결정되고, A3 변수의 값은 임의로 주어진다고 가정하자. <표 4-1>와 같은 자료가 주어졌을 때 C5.0이나 CART 등의 알고리즘은 A3변수를 뿌리마디로 선택하게 될 것이고, <그림 4-1>과 같은 최적의 의사결정나무를 찾아내지 못한다. 즉 기존의 알고리즘들은 뿌리마디에서 변수를 잘못 선택함으로서 전체자료를 정확하게 분류하는데 많은 마디를 가지는 의사결정나무를 형성하게 된다. 그러나 유전프로그래밍을 이용한 의사결정나무는 <표 4-2>의 자료를 간단하면서도 정확하게 분리해 낸다.

<표 4-1> XOR 함수 자료

A1	A2	A3	Class
T	F	T	T
T	F	F	T
F	T	F	T
F	T	T	T
F	F	F	F
F	F	F	F
T	T	T	F
T	T	F	F



<그림 4-1> GPTree를 이용한 XOR 자료 결과

4.2 실제자료

UCI Repository에 있는 실제 자료에 유전프로그래밍을 이용한 의사결정나무 형성을 시도하였다. 기존의 그리디 알고리즘과 유전프로그래밍을 이용한 의사결정나무의 직접적인 비교는 가능하지 않지만, 의사결정나무의 유효성을 판단하는 기준으로 정확한 예측력과 크기, 깊이 등을 고려하여 보았다. 기존의 그리디 알고리즘으로 CHAID, CART 등이 고려되었으며, 실제 분석에 사용된 자료는 Ionosphere, Heart-Statlog, Cleveland, Post-Operative 이다. 기존의 알고리즘은 가능하면 일반적인 파라미터 설정(default)을 사용하였다. 유전프로그래밍을 이용한 의사결정나무(GPTree)에서는 알고리즘의 효율성을 높이기 위해 LEF방법이 사용되었고, BLOAT로는 노드의 수에 대한 벌점을 부여하였다(penalty = 0.2). 그리고 모든 알고리즘에서 10-폴드(fold) 교차 타당성(cross-validation)평가를 실시하였으며 그 결과는 <표 4-2>과 같다.

<표 4-2>를 살펴보면 GPTree의 오분류율이 다른 greedy 알고리즘에 비해 비슷하거나 좋은 결과가 나타났다. Ionosphere와 Cleveland, Post-Operative 자료에서는 유전프로그래밍을 이용한

<표 4-2> 교차타당성 평가에 의한 오분류율

	CHAID	CART	GPTree
Ionosphere	0.125±0.0176	0.114±0.017	0.100±0.049
Heart-Statlog	0.211±0.025	0.193±0.024	0.211±0.063
Cleveland	0.202±0.023	0.202±0.023	0.176±0.092
Post-Operative	0.310±0.050	0.322±0.050	0.299±0.119

<표 4-3> 마디(node)의 수

	CHAID	CART	GPTree
Ionosphere	24	17	8.50±0.843
Heart-Statlog	26	29	25.60±2.867
Cleveland	19	17	28±2.237
Post-Operative	3	17	5.4±0.955

<표 4-4> 깊이(depth)

	CHAID	CART	GPTree
Ionosphere	5	5	3.50±0.71
Heart-Statlog	5	5	3.8±0.42
Cleveland	4	4	4.40±0.70
Post-Operative	2	5	2.10±0.74

의사결정나무의 오분류율이 낮은 것으로 나타났고, Heart-Statlog 자료에서는 CART 알고리즘을 이용한 의사결정나무의 오분류율이 낮은 것으로 나타났다. <표 4-3>, <표 4-4>는 전체 자료에 대한 의사결정마디의 수와 깊이를 요약한 것이다. 그 결과, Ionosphere 자료의 경우 GPTree가 다른 것들에 비해 마디의 수나 깊이에서 좋은 결과를 보였으나, Cleveland 자료의 경우 GPTree가 다른 것들에 비해 마디의 수나 깊이에서 좋지 않은 결과를 나타냈다. 앞에서 언급한 바와 같이 기

존 알고리즘과 유전프로그래밍을 이용한 의사결정나무의 직접적인 비교는 가능하지 않았지만, Ionosphere 자료의 경우 GPTree를 이용하여 상대적으로 낮은 오분류율을 가지며 깊이나 마디의 수가 적은 의사결정나무 구조를 형성함을 보았다.

5. 결론

그리디 휴리스틱(Greedy heuristic) 알고리즘을 이용한 의사결정나무 분리 방법은 일반적으로 널리 사용되고 있기는 하지만, 분리가 이루어지는 마디에서의 변수 선택이 이후의 분리에 대한 고려 없이 진행되므로 최적의 의사결정나무 형성을 보장하지 못한다. 이런 그리디 알고리즘의 한계에 대한 보완으로, 본 논문에서는 유전프로그래밍을 이용한 의사결정나무 형성을 시도해 보았다.

유전알고리즘은 복잡한 최적화 문제 해결에 효과적인 것으로 알려져 있어 기존의 결정론적 알고리즘으로 해를 찾기 어려운 큰 탐색 공간을 가진 문제에 적용된다. 유전프로그래밍은 의사결정나무 구조와 유사하고, 염색체 길이가 가변적이므로 의사결정나무를 표현하기 적당한 유전알고리즘이다.

실제로 몇 가지 자료에 적용해 본 결과, 오분류율에서는 유전프로그래밍을 이용한 의사결정나무 형성 방법이 그리디 휴리스틱 알고리즘을 이용한 방법에 비해 비슷하거나 좋은 결과를 보였다. 또한 비슷하거나 낮은 오분류율에서 유전프로그래밍을 이용한 방법이 깊이나 마디의 수가 적은 경우가 있었다. 특히 XOR 함수 자료의 경우 다른 알고리즘을 이용해서 찾아낼 수 없는 의사결정규칙을 유전프로그래밍을 이용해서 찾아낼 수 있었다. 이것은 의사결정나무의 장점 중 하나인 모형에 대한 이해와 해석의 용이함을 보장하고, 추후 분석에 있어서도 유효하게 사용될 것이다. 하지만 모든 자료에 있어 우수한 것은 아니었고, 각 알고리즘의 옵션에 따라서 결과가 달라질 수도 있을 것이다.

유전알고리즘에는 교배 확률이나 돌연변이 확률, 선택 방법이나 엘리티시즘의 사용여부 등의 다양한 설정 파라미터를 지정해 주어야 하는데, 유전프로그래밍을 이용하여 의사결정나무를 만드는 과정에서도 적절한 조정이 필요하다. 또한, 유전프로그래밍에서 일반적으로 사용되는 나무구조와는 다른, 의사결정나무라는 특정한 주제에 필요한 모집단 형성이나 유전연산자의 사용 등과 관련하여 더 많은 연구가 필요할 것이다.

참고문헌

[1] Bot, M.C.J., and Langdon, W.B. (2000). Application of Genetic Programming to Induction of Linear Classification Trees. *European Conference on Genetic Programming EuroGP2000*, Lecture Notes in Computer Science 1802, Springer, 247-258.

[2] Bot, M.C.J. (2000). Improving Induction of Linear Classification Trees with Genetic Programming. *Genetic and Evolutionary Computation Conference (GECCO) 2000.*

[3] Breiman, L. (1997). Bagging Predictors, *Machine Learning*, Vol. 24, 123-140.

[4] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees.* Belmont, CA: Wadsworth.

[5] Fu, Z. (2001). A Computational Study of Using Genetic Algorithms to Develop Intelligent Decision Trees, *IEEE.*

[6] Gathercole, C. (1998). *An Investigation of Supervised Learning in Genetic Programming,* PhD thesis, University of Edinburgh.

[7] Kass, G. V. (1980). An Exploratory Technique for Investigating Large Quantiles of Categorical Data, *Applied Statistics,*29, 119-127.

[8] Koza, J. R. (1991). Concept formation and decision tree induction using the genetic programming paradigm. In Schwefel, Hans-Paul, and Maenner, Reinhard (editors). *Parallel Problem Solving from Nature.* Berlin: Springer-Verlag. 124-128.

[9] Koza, J. R. (1992). *Genetic Programming.* MIT Press.

[10] Loh, W.-Y. and Shih, Y.-S. (1997), Split Selection Methods for Classification Trees, *Statistica Sinica,* 7, 815-840.

[11] Nikolaev, N. I., and Slavov, V. (1997) Inductive Genetic Programming with Decision Trees. *In 9th European Conference on Machine Learning,* Plague, Czech Republic, 3-26.

[12] Papagelis, A, and Kalles, D. (2000). GATree: Genetically Evolved Decision Trees, IEEE.

[13] Papagelis, A, and Kalles. D. (2001). Breeding Decision Trees Using Evolutionary Techniques. *ICML.*

[14] Quinlan, J. R. (1979). Discovering rules by induction from large collections of examples. In D. MIchie (ed.), *Expert Systems in the Mirco Electronic Age.* Edinburgh, UK: Edinburgh University Press.

[15] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning 1,* 1, 81-106.

[16] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning,* Morgan Kaufmann Publishers, CA.

[2003년 8월 접수, 2003년 10월 채택]