經營科學
第23卷 第1號
2006年 5月

순차적으로 선택된 특성과 유전 프로그래밍을 이용한 결정나무

김효중* · 박종선*

A Decision Tree Induction using Genetic Programming with Sequentially Selected Features

Hyojung Kim* · Chongsun Park*

■ Abstract ■

Decision tree induction algorithm is one of the most widely used methods in classification problems. However, they could be trapped into a local minimum and have no reasonable means to escape from it if tree algorithm uses top-down search algorithm. Further, if irrelevant or redundant features are included in the data set, tree algorithms produces trees that are less accurate than those from the data set with only relevant features.

We propose a hybrid algorithm to generate decision tree that uses genetic programming with sequentially selected features. Correlation-based Feature Selection (CFS) method is adopted to find relevant features which are fed to genetic programming sequentially to find optimal trees at each iteration.

The new proposed algorithm produce simpler and more understandable decision trees as compared with other decision trees and it is also effective in producing similar or better trees with relatively smaller set of features in the view of cross-validation accuracy.

Keyword : Decision Tree, Correlation based Feature Selection, Genetic Programing

논문접수일 : 2005년 6월 10일 논문게재확정일 : 2005년 12월 13일
* 성균관대학교 경제학부

1. 서 론

결정나무는 분류 또는 예측의 과정이 비교적 간단한 나무구조에 의해 추론 규칙(induction tule)을 표현하기 때문에 데이터 마이닝 분야에서 많이 사용되며, 분류 또는 예측하고자 하는 대상인 목표변수의 성격에 따라 분류나무(classification tree)와 회귀나무(regression tree)로 구분되어진다[4, 21, 25].

결정나무는 사례들 사이의 동질성, 연관성이 최대화되도록 분류하는 특성을 선택하여 자료들을 각각 동질적인 그룹으로 나누도록 하는 알고리즘으로 구성되어지며, 과정을 끝마디(terminal node 또는 leaf)에 속하는 모든 사례들이 같은 클래스 값을 가짐으로써 분류가 불가능할 경우 또는 분류를 중지하는 정지 규칙(stopping rule)을 만족할 때까지 반복함으로써 결정나무를 생성하게 된다. 카이제곱 통계량을 분리기준으로 하는 CHAID, 지니 지수(gini index)를 분리기준으로 사용하는 CART와 Quest, 그리고 엔트로피 지수(entropy index)를 분리기준으로 사용하는 C4.5(또는 J4.8)등과 같은 다양한 결정나무 생성 알고리즘들이 연구되어졌다.

그러나 기존의 결정나무 생성 알고리즘은 잉여특성(redundant feature)이 존재할 경우, 결정나무의 크기를 증가시켜 해석하기 어려운 결정나무를 생성하게 될 가능성이 높아지게 될 뿐 아니라, 결정나무의 전체적인 정확성(overall accuracy)을 손상시키게 된다[10]. 따라서 자료에서 가능한 한 부적절하거나 잉여인 특성들을 되도록 많이 제거하거나 식별하는 특성 선택(feature selection)을 수행함으로써 자료의 차원을 축소하고, 학습 알고리즘의 운용(operation)을 보다 효과적이고 빠르게 할 수 있다.

특성 선택 알고리즘은 일반적으로 두 가지 큰 범주-추출 접근(filter approaches)과 보자기 접근(wrapper approaches)-로 나눌 수 있다. 추출 접근은 특성들과 추론 알고리즘에 의한 분류자를 서로 독립적인 것으로 고려하고, 정보 이득(information gain), 교차-엔트로피(cross-entropy)등 통계적 혹은 정보 이론적 척도들을 특성의 가중값(weight)으로 사용하여 목표변수(또는 클래스)와 특성의 관계를 포착하기 때문에 일반적으로 특성 선택 속도가 빠르며 고차원의 자료에서 사용 가능하다는 장점을 가진다[8, 13, 14]. 한편 특성 선택에 대한 보자기 접근은 특성 부분집합의 가치를 추정하기 위해 추론 알고리즘(induction algorithm)을 사용하여 특성들의 공간 탐색을 수행함으로써, 훈련 자료와 추론 알고리즘 사이의 특정한 상호작용이 밝혀진다는 사실 때문에 추출 접근의 결과보다 더 좋은 결과를 나타내지만, 추론 알고리즘이 반복적으로 호출되기 때문에 추출 접근보다 실행 속도 면에서 느린 경향이 있다[1, 11].

기존의 결정나무 생성 방법들은 사례들을 분류하는 각 분류마디에서 최적인 특성을 선택하여 사례들을 분류하기 때문에 분류마디에서는 최적이지만, 전체적인 결정나무에서는 최적이 되지 않는 국소 최적(local optima)의 문제가 발생할 수 있다[18]. 기존의 결정나무 생성 알고리즘이 가지고 있는 국소 최적의 문제에 빠지지 않고 전역 최적에 가까운 결정나무를 생성할 가능성이 높은 방법으로 유전 프로그래밍(genetic programming)에 의한 결정나무 생성 알고리즘이 알려져 있다[3, 7, 15, 19]. 그러나 유전 프로그래밍에 의한 결정나무 생성 방법은 특성의 수가 많은 경우 시간적, 계산적 복잡성이 증가하는 문제가 발생한다.

대용량 자료에 대한 효율적인 결정나무를 생성하기 위해서는 결정나무 생성에 불필요한 특성들을 제거하는 것과 국소 최적에 도달하게 되는 것을 방지하는 것이 요구되어진다. 본 논문에서는 MDI (Measure of Departure from Independence)를 이용하여 특성과 특성 그리고 특성들과 클래스의 연관성을 동시에 고려함으로써 결정나무 생성에 필요한 특성들을 선택하는 Hall의 상관에 의한 특성 선택(Correlation based Feature Selection ; CFS)을 이용하여 결정나무 생성에 필요한 적은 수의 특성들을 선택하고, 그 특성들을 순차적으로 유전 프로그래밍에 의한 결정나무 생성 알고리즘에 이용

하는 혼합 알고리즘(hybrid algorithm)을 고려함으로써 기존의 결정나무 생성 알고리즘들이 가지는 한계(정확성의 손상과 국소최적)를 극복하는 것에 대해 연구하고자 한다.

본 논문은 먼저, 제2장에서는 특성 선택을 살펴보고, 제3장에서는 유전 프로그래밍을 이용한 결정나무 생성에 대해 살펴보도록 할 것이며, 제4장에서는 특성 선택과 유전 프로그래밍 결정나무를 이용한 제안 알고리즘에 대해 살펴보도록 한다. 제5장에서는 실제 자료에 대한 제안된 알고리즘과 기존 알고리즘들의 비교를 수행할 것이며, 마지막으로 제6장에서는 본 논문의 결론에 대하여 언급하도록 한다.

2. 특성 선택

본 절에서는 결정나무 생성에 효과적인 적은 수의 특성을 선택하는 방법에 대해 살펴보도록 한다. 특성 선택 알고리즘은 일반적으로 추출 접근(filter approaches)과 보자기 접근(wrapper approaches)의 두 가지 큰 범주로 나눌 수 있다. 추출 접근 방법들은 학습 알고리즘과 독립적으로 운용되는 것으로 바람직하지 못한 특성들은 추론(induction)이 시작되기 전에 자료로부터 추출하는 접근 방법이다[2, 8, 9, 12, 14, 20, 22]. 이에 반해 보자기 접근 방법들은 실제 추론 방법을 부-함수(subroutine)처럼 사용하여 특성들을 선택하는데, 특성들의 정확도를 추정하기 위해 통계적 재-추출(re-sampling)과 같은 교차 타당성(cross validation)을 이용하는 접근 방법이다[5, 6, 11, 24].

Hall[8]은 추출 접근 방법인 상관에 의한 특성 선택을 제안하였다. CFS 방법은 명목형 클래스에 대한 감독된 학습에 필요한 특성들을 선택하기 위하여 모든 특성들과 클래스를 이산화 과정에 의해 동일한 형식(즉, 명목형)으로 처리하였으며, 평가함수에 사용되는 특성과 특성, 특성과 클래스 사이의 상관을 측정하기 위해 대칭적 불확실성(symmetrical uncertainty), 최소 표현길이(Minimum Description Length : MDL), Relief 등의 3가지 척도들 중 한 가지를 선택하여 사용하는 것을 고려하였다.

Hall은 3가지 척도들의 표준화되고 평균화된 값들을 이용하여, 다음과 같이 각 특성들의 특성 평가 함수의 값을 구하였다.

$$M_s = (k \cdot \overline{r_{cf}}) / (k + k(k-1) \cdot \overline{r_{ff}})^{1/2}$$

여기서, M_S는 k개 특성들을 포함하는 특성 부분집합 S의 발견적 "가치(merit)"이며, $\overline{r_{cf}}$는 특성과 클래스 상관의 평균 ($f \in S$), 그리고 $\overline{r_{ff}}$는 특성과 특성간의 내부 상관의 평균이다.

구해진 M_S값 중에서 가장 큰 값을 가지는 특성을 선택한 후, 전진 선택 탐색을 통해 선택된 특성을 포함하는 집합에 순차적(sequential)으로 특성을 추가하였다. 선택된 하나의 특성을 포함하는 두 개의 특성에 대하여 다시 특성 평가 함수를 이용하여 특성을 추가로 선택하는 것을 반복한다. 특성의 추가를 반복적으로 수행했을 때 추가로 포함되는 특성의 가치가 증가하지 않거나 미세하게 증가하는 경우, 특성이 추가되기 이전까지의 특성 부분집합을 선택하도록 하였다. 그는 CFS 방법에 의해 선택된 특성들의 부분집합이 결정나무 생성에서 효과적인 역할을 수행한다는 것을 보였다. 그러나 CFS 방법은 특성 평가 함수를 계산하기 위한 척도로 사용되는 상관(연관)을 측정하기 위해 모든 특성과 클래스들을 이산화(discretization) 과정을 통해 명목형으로 처리하여야 하는 단점을 가진다.

Lee 등[17]은 이산화 과정을 거치지 않고 명목형 특성들과 연속형 특성들로 이루어진 혼합 자료에서의 독립성 검정에 의한 연관성 측정을 위해 MDI(Measure of Departure from Independence)를 제안하였다. MDI는 독립성 검정을 수행하는 데 사용되는 검정 통계량의 유의 확률인 p-값을 사용하여 혼합 자료에서의 특성들과 클래스 사이의 연관성을 측정하였다. 따라서 혼합 자료에서의 특성 선택을 위해 이산화의 과정이 필요하지 않은 MDI를 특성 평가함수의 척도로 이용한 CFS 방법을 고려할 수 있을 것이다. MDI의 $p-$ 값은 다음과 같이

계산되어진다.

i) X, Y가 모두 명목형일 경우(피어슨 X^2통계량의 $p-$값)

$$p-\text{값} = P(\chi^2_{(r-1)(c-1)} > \sum_{i=1}^{r}\sum_{j=1}^{c}(O_{ij}-E_{ij})^2/E_{ij})$$

여기서, O_{ij}는 X의 i번째와 Y의 j번째에서의 관측 빈도이고, E_{ij}는 X의 i번째와 Y의 j번째에서 기대 빈도이다.

ii) X는 명목형이고, Y는 연속형일 경우(크루스칼-왈리스 X^2통계량의 $p-$값)

$$p-\text{값} = P(\chi^2_{k-1} > 12/N(N+1)\cdot \sum_{i=1}^{k} n_i\cdot[\overline{R}_i-(N+1)/2]^2)$$

여기서, $N=n_1+n_2+\cdots+n_k$이고, n_i는 X의 i번째 범주의 자료의 수 그리고, $\overline{R}_i$는 X의 i번째 범주에서의 Y들의 순위 평균이다.

iii) X, Y가 모두 연속형일 경우(스피어만 r_s통계량의 $p-$값)

$$p-\text{값} = P(r_{s(n)} > 1-[6/n(n^2-1)\cdot\sum_{i=1}^{n}(R_i-S_i)^2])$$

여기서, R_i는 X의 i번째 자료의 순위이고, S_i는 Y의 i번째 자료의 순위를 나타내며, $r_{s(n)}$는 자료의 수 n에 해당하는 스피어만 순위상관계수 분포의 임계값이다.

따라서 본 논문에서 사용되는 특성 평가 함수의 값은

$$M_S=[k\cdot\overline{p_{cf}}]/[k+k(k-1)\cdot\overline{p_{ff}}]^{1/2}$$

여기서, $\overline{p_{cf}}$는 특성과 클래스의 MDI p-값의 평균이고, $\overline{p_{ff}}$는 특성과 특성의 MDI p-값의 평균이다.

로 구해질 수 있을 것이다.

3. 유전 프로그래밍에 의한 결정나무 생성

유전 프로그래밍은 주어진 문제를 풀기 위해 유전자 알고리즘을 이용하여 컴퓨터 프로그램인 개체군을 생성, 진화시킨다. 유전 프로그래밍에서 각 개체(염색체)는 주어진 문제 영역에 적합한 함수와 터미널의 조합이다. 함수 집합은 산술 연산자나 수학함수, 논리 연산자 등을 포함하며, 터미널 집합은 문제에 적합한 입력 자료(input data)의 값이나 다양한 상수를 포함한다[16]. 유전 프로그래밍을 이용하여 결정나무를 형성하기 위해서는 연속형 특성인 경우의 분리 기준을 결정하는 문제와 최대 깊이, 초기 개체군의 수, 그리고 기타 교배나 돌연변이 확률 등과 같은 모수를 결정하여야 한다. 본 논문에서는 유전 프로그래밍에 의한 결정나무를 생성하기 위해 Qureshi의 GP system(GPsys[1])을

〈표 1〉 GPsys의 표준 설정

목표	모든 훈련사례를 정확하게 분류
터미널 집합 (Terminal set)	특성, 상수, 분류값
함수 집합 (Function set)	분류 조건(CheckCondition)
선택(Selection)	토너멘트 선택(크기=7)
모수(Parameters)	모집단 크기 : 200 반복 횟수 : 1 세대변이(generation) 수 : 100 엘리티즘 사용 교배 확률 : 0.9 돌연변이 확률 : 0.1
초기 모집단 형성	RAMPED_HALF_AND_HALF[2]
중단 조건	모든 사례들이 정확하게 분류될 때

1) http://www.cs.ucl.ac.uk/external/A.Qureshi/gpsys_doc.html

2) 초기 모집단을 형성할 때 50%는 모든 잎마디에서 지정된 깊이의 나무구조가 형성되고, 나머지 50%는 지정된 깊이내에서 임의로 나무구조가 형성되도록 초기 모집단을 형성하는 방법.

수정한 Bot의 프로그램을 이용하였다. GPsys는 자바(Java) 언어를 이용해 구현한 것으로 토너멘트 선택에 의해 선택된 유전자를 다음 세대에 그대로 보존시키고, 나머지 선택되지 않은 유전자들을 다시 무작위로 생성시키는 엘리티즘(elitism)을 사용하고 있다. GPsys의 표준 설정은 <표 1>과 같다.

4. 특성 선택과 유전 프로그래밍을 이용한 결정나무

결정나무를 생성하기 위해 필요한 특성들을 특성 선택에 의해 선택한 후, 그 특성들을 이용한 유전 프로그래밍 결정나무를 생성하기 위한 알고리즘은 <표 2>와 같다.

<표 2>의 알고리즘을 각 단계별로 살펴보면, 단계 1에서는 MDI 방법에 의한 p-값을 계산하게 된다. 이때 p-값은 특성과 특성, 특성과 클래스의 자료 형태에 따라 다르게 계산된다. 특히 명목형-명목형의 자료 형태에 대한 피어슨 X^2통계량의 p-값 계산은 사례의 수에 의해 크게 영향을 받는다. 사례의 수가 많은 경우 일반적인 χ^2-검정 방법에 의한 근사적 p-값이 사용될 수 있지만, 자료의 수가 적은 경우는 근사적 p-값은 연관성을 나타내기 위한 적절한 값이 되지 않기 때문에 몬테칼로(monte carlo)법에 의해 계산된 p-값을 이용한다.

단계 2에서 MDI의 p-값을 특성과 특성, 특성과 클래스의 연관성을 측정하기 위한 척도로 사용하여 특성 평가 함수를 계산한다. 계산된 값은 p-값의 영향을 받게 되는데, 만약 특성과 특성, 특성과 클래스 사이의 연관성이 매우 높을 경우 p-값은 아주 작은 값을 가지게 된다. 따라서 결정나무 생성에 가장 필요한 특성은 매우 작은 p-값을 가지게 되므로, 가장 작은 가치(merit)를 가진 특성부터 순서대로 특성들을 나열한다.

단계 3에서는 단계 2에 의해 나열된 특성들을 순차적으로 유전 프로그래밍(Genetic Programming; GP)에 입력하여 최적의 해를 찾는다. 특성 선택에 의해 처음으로 선택된 하나의 특성은 GP에 의해 사례들을 최적 분리(훈련 집합의 정확도)에 의해 분류한다. 이후의 특성 추가에서 여러 가지 가능한 특성 조합이 가능할 것이며, GP에 의한 결정나무 생성은 가능한 특성들의 조합을 모두 고려함으로써 전역 최적에 가까운 결정나무를 생성할 수 있는 가능성을 높일 수 있다. 나열된 특성의 일부를 모두 GP에 입력하여 결정나무를 생성하면 입력된 특성들만의 국소 최적 결정나무를 생성하게 되므로 결정나무 생성에 필요한 특성을 적절하게 판단할 수 없게 된다. 따라서 전진 선택과 같은 순차적인 특성 추가에 의해 GP에 특성들을 입력하는 것이 적절할 것이다.

〈표 2〉 CFS 방법에 의해 선택된 특성을 이용한 유전 프로그래밍 결정나무 생성 알고리즘

단계 1 : MDI를 이용하여 특성과 클래스, 특성과 특성에 대한 p-값을 구한다.

단계 2 : 특성 평가 함수

$$M_S = [k \cdot \overline{p_{cf}}]/[k + k(k-1) \cdot \overline{p_{ff}}]^{1/2}$$

여기서, $\overline{p_{cf}}$는 특성과 클래스의 MDI p-값의 평균이고, $\overline{p_{ff}}$는 특성과 특성의 MDI p-값의 평균이다.

를 계산하여 값이 작은 것부터 순서대로 나열한다.

단계 3 : 나열된 특성들에서 순차적으로 하나의 특성을 선택하여 다음을 반복수행.

Ⅰ. 선택된 특성을 이용해 유전 프로그래밍 결정나무 방법에 의해 결정나무를 생성하는 조합을 구하고 그에 해당하는 훈련집합 정확도와 교차 타당성을 구한다.

Ⅱ. 교차 타당성이 증가하지 않거나 미세하게 증가하는 경우 중지.

다음으로 특성 추가를 중지해야 하는 중단 임계는 생성된 결정나무의 여러 평가 기준-나무의 크기(size of the tree), 교차 타당성(cross validation) 등-에 의해 설정되어질 수 있다. 보통의 유전 프로그래밍에서는 최적을 판단하기 위한 기준으로 목적 함수(적합도)를 사용할 수 있지만, 제안된 방법에서는 그러한 기준을 사용할 수 없다. 만약 적합도를 기준으로 할 경우, C4.5등과 마찬가지로 주어

진 특성들을 이용해 모든 사례를 가장 잘 분류할 수 있는 결정나무 생성 조합을 찾기 위해 나무의 크기를 계속 확대시키는 문제(부풀림[Bloat] 문제)가 발생하기 때문이다[16]. 따라서 나무의 크기를 확대시킴으로써 발생하는 과대적합의 영향이 비교적 적은 척도들을 중단 임계를 결정하기 위한 목적함수로 사용함으로써 부풀림 문제를 방지할 수 있을 것이다. 본 논문에서는 중단 임계를 결정하기 위한 척도로 교차 타당성(cross validation)을 사용하였다. 따라서 교차 타당성이 감소 또는 변화량이 작을 때까지 특성을 순차적으로 추가하여 결정나무를 생성하였다. 예를 들어 3개의 특성(A, B, C)을 이용한 결정나무를 생성할 경우, 특성 평가함수에 의해 B, C, A의 순서로 작은 가치를 갖는다면, 특성 $\{B\}, \{B, C\}, \{B, C, A\}$가 GP에 포함된다. 포함된 각 특성들에 대하여 사례들을 훈련집합의 정확도가 높은 결정나무를 선택하고, 그것에 대한 교차 타당성을 구한다. 구해진 교차 타당성이 증가하지 않거나, 미세한 증가를 나타낼 경우, 바로 직전의 특성 집합을 이용한 결정나무를 최적 결정나무로 선택한다.

그러나 유전 프로그래밍은 교배에 의해 세대변이를 하기 때문에 같은 교차 타당성을 가지는 여러 가지 결정나무 생성 조합이 존재하게 된다. 따라서 보다 작은 결정나무를 생성하기 위한 제약이 필요하다. Soule[23]은 유전 프로그래밍의 각 마디(node) 또는 깊이(depth)에 페널티(penalty)를 부과한 적합도 함수를 수정하여 부풀림을 줄이는 것을 제안하였다. 즉, 부과된 페널티에 따라 생성되는 결정나무의 형태가 달라진다. 따라서 너무 큰 페널티를 부여하면 클래스 값 자체로만 사례를 분류하게 되고, 페널티를 부여하지 않으면 잉여 마디가 포함된 과적합을 발생시킨다. Bot 등[3]은 부풀림을 피하기 위한 여러 가지 척도들을 비교하였으며, 그 결과 마디 패널티가 0.5이고 깊이 패널티가 2.0일 때 최적의 유전 프로그래밍 결정나무를 생성한다는 것을 보였다. 그러나 본 논문에서는 특성 선택에 의한 결정 나무들의 비교를 위해 가급적 패널티의 영향을 최소화하는 것이 필요하다. 따라서 마디 패널티는 아주 미세한 값을 사용하고 깊이 패널티는 주지 않았다.

5. 실제 자료를 이용한 비교

이 절에서는 몇 가지 실제 자료(UCI 저장소)에 대하여 잘 알려진 결정나무 생성 알고리즘인 C4.5(이하 J4.8 결정나무), Hall의 CFS 방법에 의한 J4.8 결정나무(이하 CFS 결정나무), 그리고 특성선택을 이용한 유전 프로그래밍 결정나무(이하 GP 결정나무)의 결과를 비교하도록 한다. J4.8, CFS 방법은 Weka(version 3.4)를 이용하였으며, 제안방법은 R(version 7.1.1)을 이용하여 MDI p-값을 계산한 후 수정된 CFS 방법에 의해 특성을 선택하였다. 그리고 GP 결정나무는 Gpsys와 Bot의 프로그램을 이용하여 구현하였다. 또한 CFS 결정나무와 J4.8 결정나무에 대한 교차 타당성은 Weka의 Experimenter를 이용하여 10번의 반복 실험 후의 평균값을 사용하였으며, GP 결정나무는 무작위로 선택된 10개의 난수(random number)를 이용하여 구한 교차 타당성의 평균값을 사용하였다. 이때 GP 결정나무는 현실적으로 가지치기를 적용하기 어렵기 때문에 GP 결정나무와 CFS 결정나무의 교차 타당성의 비교를 위해 CFS 결정나무는 가지치기를 수행하지 않았을 때의 결과를 사용한다. 한편, J4.8 결정나무는 GP(또는 CFS) 결정나무의 상대적인 측면을 살펴보기 위한 것이므로 가지치기를 수행한 결과를 이용하도록 한다.

GP 결정나무는 주어진 자료를 이용하여 MDI p-값을 구하고, 그 값을 이용해 CFS 방법으로 선택할 특성들의 순서를 정하였다. 이때 특성 값과 클래스 값은 GPsys를 사용하기 위해 모두 숫자로 변환하였다.

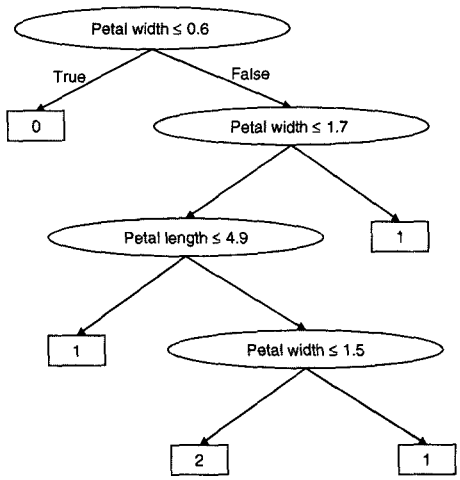
$$\text{변환된 자료값} = \text{상수} + \frac{(\text{자료값} - \text{해당특성의 최소값})}{(\text{해당특성의 최대값} - \text{해당특성의 최소값})}$$

명목형 특성인 경우 가능한 분리값의 수는 특성이 포함하고 있는 인수의 수로 설정한다. 즉, 명목의 수가 2인 경우는 이진 분리(binary split)를 수행하는 결정나무를 생성하게 되고, 명목의 수가 3 이상인 경우는 다중 분리(multiple split)를 수행하는 결정나무를 생성하도록 설정한다. 다중 분리를 수행할 경우, 결정나무의 크기가 복잡한 결과를 나타낼 수도 있지만, 자료의 변환을 최소로 하여 정보의 손실을 최소화할 수 있을 것으로 생각되어진다. 연속형 특성인 경우 인수의 수를 2(즉, 지분되는 마디의 수가 2개)로 설정하였다. 그리고 상수는 명목형 특성의 최대 특성값보다 큰 정수 값을 사용한다. 또한 연속형 특성의 분리 기준값은 변환된 자료의 최소값과 최대값 사이의 임의의 실수로 지정된다.

5.1 붓꽃 자료(Iris Data)

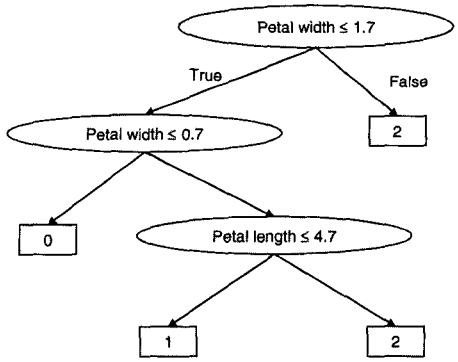
붓꽃 자료는 Sepal length, Sepal width, Petal length, Petal width 등의 특성들과 1개의 클래스(type)로 이루어진 150개의 사례를 포함하고 있다. 4개의 특성은 모두 연속형 특성이며, 클래스는 3개의 값(Iris Setosa, Iris Versicolour, Iris Virginica)을 갖는다.

먼저, J4.8에 의하면 모든 특성을 사용하여 <그림 1a>와 같이 나무 크기(size of the tree)가 9인 결정나무가 생성되어지게 된다. 10층 교차 타당성 실험을 10번 반복한 결과 평균적으로 94.73%의 정확도를 나타내었다. CFS 방법은 Weka의 이산화 알고리즘에 의해 자료를 이산화 한 후, 전진 선택 방법에 의해 Petal length과 Petal width를 특성으로 선택하였으며 특성 집합의 가치(merit)는 0.887을 나타내었다. CFS방법에 의해 선택된 두 특성만을 사용하였을 경우, CFS 결정나무는 J4.8의 결과(<그림 1a>)와 같은 결과를 나타내었으며, 10층 교차 타당성 실험을 10회 반복한 결과 평균적으로 94.80%의 정확도로 J4.8의 결과와 유사한 정확도를 나타내었다.



여기서, 0 : Iris Setosa, 1 : Iris Versicolor, 2 : Iris Virginica

〈그림 1a〉 J4.8 결정나무(붓꽃 자료)



여기서, 0 : Iris Setosa, 1 : Iris Versicolor, 2 : Iris Virginica

〈그림 1b〉 GP 결정나무(붓꽃 자료)

MDI p-값과 CFS 방법에 의해 Petal width, Petal length, Sepal length, Sepal width의 순서로 특성이 선택되었으며, 특성의 수가 1, 2, 3, 4인 경우에 <표 3>과 같은 교차 타당성 결과를 얻을 수 있다.

〈표 3〉 붓꽃 자료에서의 특성 수에 따른 GP 결정나무의 결과

	FN 1	FN 2	FN 3	FN 4
훈련 집합 정확도	0.96	0.980	0.973	0.987
교차 타당성(1회)	0.933	0.953	0.940	0.953

주) FN i : i개의 특성을 사용

〈표 4〉 붓꽃 자료에 대한 J4.8, CFS, GP 방법의 비교(10층 교차 타당성 실험 10회 반복)

결정나무	나무 크기	사용된 특성의 수	훈련 자료의 정확도(%)	교차 타당성(%) ± 표준 오차
J4.8	9	2	98.00%(147/150)	94.73% ± 5.30%
CFS	9	2	98.00%(147/150)	94.80% ± 5.24%
GP	7	2	98.00%(147/150)	93.74% ± 0.19%

특성 수의 변화에도 불구하고 교차 타당성 정확도는 거의 변화(약 2%정도의 차이)가 없는 것으로 나타났다. 또한 평균 정확도도 거의 차이가 없는 것으로 나타났다. 따라서 2개의 특성을 선택하여 GP 결정나무를 생성하는 것이 적절하다고 판단된다. GP 방법에 의해 생성된 결정나무는 나무 크기가 7인 <그림 1b>와 같은 결정나무 조합을 나타내었다.

GP 방법에 의해 생성된 최적 결정나무는 10층 교차 타당성 실험을 10회 반복하면 평균적으로 교차 타당성의 93.74%를 나타내었다. 세 가지 방법에 의해 생성된 결정나무에는 2가지 특성(petal width와 petal length)이 공통적으로 사용되었다. 세 가지 방법의 결과는 <표 4>와 같다.

5.2 심장병 자료(Heart Disease Data)

심장병 자료는 UCI 저장소(UCI repository)의 자료로 전체 75개 특성 중 나이 성별 등을 포함한 13개 특성을 이용하여 심장병 존재 여부를 판단하는 270개 사례를 포함하고 있으며 결측값은 없다. 13개 특성은 6개의 연속형 특성(1, 4, 5, 8, 10, 12번째 특성)과 7개의 명목형 특성(2, 3, 6, 7, 9, 11, 13번째 특성)으로 이루어져 있으며, 클래스는 심장병 여부를 나타내는 2개의 값(absent, present)을 가진다.

먼저, J4.8 결정나무는 9개의 특성[age, chest (chest pain type), press(resting blood pressure), electro(resting electrocardiographic result), rate (maximum heart rate achieved), vessel(number of major vessel colored by flourosopy), slope(the slope of the peak exercise ST segment), thal(3 ; normal, 6 ; fixed defect, 7 : reversable defect), oldpeak(oldpeak=ST depression induced by exercise relative to rest)]을 사용하여 결정나무를 생성하며 나무 크기는 41이다. 모든 자료를 훈련 자료로 이용할 경우 J4.8의 정확도는 91.85%이며, 10층 교차 타당성 실험을 10회 반복한 평균 정확도는 76.93%를 나타내었다. CFS 방법은 전진 선택에 의해 6개의 특성(chest, rate, angina, vessel, thal, oldpeak)을 선택하게 된다. 또한 이때의 특성 부분집합의 최고 가치는 0.316을 나타내었다. CFS 결정나무는 나무 크기가 21인 결정나무가 생성되게 된다. 모든 자료를 훈련 자료로 이용할 경우 CFS 결정나무의 정확도는 88.89%이었으며, 10층 교차 타당성 실험을 10회 반복하면 평균적으로 79.63%의 정확도를 나타내었다.

MDI p-값과 CFS 방법에 의하면 thal, vessel, chest, rate, angina, oldpeak, slope, sex, age, serum, electro, press, sugar의 특성이 순서대로 선택되어진다. <표 5>는 특성 수의 변화에 따른 교차 타당성 결과를 나타내고 있다. <표 5>를 살펴보면, 특성의 수가 3인 경우(FN3)에서 가장 높은 교차 타당성을 나타내고 있다. 특성의 수가 증가함에 따라 훈련 집합의 정확도도 함께 증가하고 있는 것을 알 수 있다. 그러나 그러한 정확도의 증가는 특성 수의 증가에 의한 결과이며, 보다 많은 특성이 사용됨에 따라 마디(node)의 수와 깊이도 함께 증가하여 보다 복잡하고 난해한 결정나무를 생성하게 된다. 3개의 특성(thal, chest, vessel)을 이용한 제안된 GP방법에 의한 결정나무는 나무 크기가 18인 결정나무를 생성한다.

<표 6>을 살펴보면, GP 방법의 훈련 자료 정확도는 86.30%이며, 10층 교차 타당성 실험을 10회 반복한 정확도는 84.80%를 나타내었다.

〈표 5〉 심장병 자료에서의 특성 수 변화에 따른 GP 결정나무 결과

	FN 1	FN 2	FN 3	FN 4	FN 5	FN 6	FN 7
훈련 자료 정확도	0.763	0.796	0.863	0.874	0.874	0.915	0.904
교차 타당성(1회)	0.759	0.785	0.844	0.833	0.822	0.800	0.811

주) FN i : i개의 특성을 사용

〈표 6〉 심장병 자료에 대한 J4.8, CFS, GP 방법의 비교(10층 교차 타당성 실험 10회 반복)

결정나무	나무 크기	사용된 특성의 수	훈련 자료의 정확도(%)	교차 타당성(%) ± 표준 오차
J4.8	41	9	91.85%(248/270)	76.93% ± 7.31%
CFS	28	6	88.89%(240/270)	79.63% ± 6.73%
GP	22	3	86.30%(233/270)	84.40% ± 0.68%

〈표 7〉 독일 신용 자료에서의 특성 수 변화에 따른 GP 결정나무 결과

	FN 1	FN 2	FN 3	FN 4	FN 5	FN 6
훈련 자료 정확도	0.700	0.723	0.769	0.770	0.792	0.780
교차 타당성(1회)	0.676	0.714	0.730	0.724	0.710	0.707

주) FN i : i개의 특성을 사용

〈표 8〉 독일 신용 자료에 대한 J4.8, CFS, GP 방법의 비교(10층 교차타당성 실험 10회 반복)

결정나무	나무 크기	사용된 특성의 수	훈련 자료의 정확도(%)	교차 타당성(%) ± 표준 오차
J4.8	140	17	85.20%(852/1000)	71.25% ± 3.17%
CFS	30	3	75.40%(754/1000)	71.61% ± 3.25%
GP	44	3	76.80%(768/1000)	73.68% ± 0.39%

5.3 독일 신용 자료(Germany Credit Data)

독일 신용자료는 UCI 저장소(UCI repository)의 자료로 전체 20개의 특성을 이용하여 고객의 신용 등급(good 또는 bad)을 판단하는 1,000개의 사례를 포함하고 있으며 결측값은 없다. 20개의 특성은 7개의 연속형 특성(duration, amount, installp, resident, age, existcr, depends)와 13개의 명목형 특성으로 이루어져 있으며, 클래스는 고객 신용 등급을 나타내는 2개의 값(good, bad)으로 이루어져 있다.

먼저 J4.8 결정 나무는 가지치기를 수행한 후에도 나무의 크기가 140인 결정 나무를 생성하게 되며, 이때 사용되는 특성은 17개의 특성(20개 특성 중 installp, coapp, depends를 제외한 나머지 특성들)을 사용하고 있다. 자료들을 훈련 자료로 사용하였을 경우 정확도는 85.5%를 나타내었다. 10층 교차 타당성 실험을 10번 반복했을 경우 평균적으로 71.25%의 정확도를 나타내었다. CFS 결정나무는 20개의 특성 중에서 3개의 특성(checking, duration, history)을 선택하였으며, 이때의 최고 가치는 0.076이었다. CFS 결정 나무 크기는 30이었다. 또한 모든 자료를 훈련 자료로 사용할 경우 정확도는 75.4%를 나타내었다. 10층 교차 타당성 실험을 10번 반복한 결과 평균적으로 72.27%의 정확도를 나타내었다. 마지막으로 MDI p값과 CFS 방법에 의하면 checking, history, duration, saving, property, housing, purpose, …의 특성이 순서대로 선택되었으며, 특성 수 변화에 따른 GP 결정 나무의

교차 타당성은 <표 7>과 같다.

<표 8>에서 3개의 특성이 사용된 경우(FN3)에서 가장 높으므로, 3개의 특성을 사용한 GP 결정나무를 생성하는 것이 적절할 것으로 판단된다. 따라서 3개의 특성을 사용할 경우, GP 결정 나무는 나무 크기(tree size)가 44로 CFS 결정 나무에 비해 크게 나타나고 있으며 이때의 정확도는 76.80% 정도를 나타내었다. 또한 10층 교차 타당성 실험을 10번 반복한 결과 평균적으로 73.68%정도의 정확도를 나타내었다.

6. 결 론

기존의 결정나무 생성 알고리즘들의 문제를 해결하는 방안으로 MDI p-값을 특성과 특성, 특성과 클래스의 상관(연관성)을 측정하는 척도로 이용한 CFS 방법을 사용하여 특성들에 대한 가치를 평가하고, 가치가 결정된 변수들을 전진 선택 방법으로 유전 프로그래밍에 순차적으로 포함시켜 결정나무를 생성하는 알고리즘에 대하여 살펴보았다.

제안된 알고리즘과 기존 알고리즘의 비교 결과 상대적으로 적은 수의 특성을 사용함으로써 보다 간결하고 이해하기 쉬운 결정나무를 생성하였으며, 교차 타당성은 비슷하거나 더 좋은 결과를 나타내었다. 또한 CFS 결정나무와 GP 결정나무를 비교했을 때, 동일한 수의 특성이 사용될 경우, 제안된 방법의 결정나무가 CFS 결정나무보다 크기가 더 작은 것으로 나타났으며, 동일한 수의 특성이 사용될 경우 제안된 방법의 교차 타당성이 CFS 방법의 교차 타당성과 비슷하거나 더 좋은 결과를 나타내었다.

결과적으로 제안된 알고리즘은 J4.8 알고리즘과 CFS에 의한 J4.8 알고리즘의 중간 정도의 성능을 나타낸다고 할 수 있을 것으로 생각된다. 따라서 제안 알고리즘은 일반적인 결정나무 알고리즘에 비해 특성 선택을 이용하여 예측력을 많이 떨어뜨리지 않으면서 복잡하지 않은 결정나무의 생성을 가능하게 하며, 유전 프로그래밍을 사용하여 효율성을 향상시킬 수 있을 것으로 생각되어진다.

향후 본 논문에서 고려하지 않았던 연속형 특성의 다중 이산화(multi-values discretization)와 생성된 결정나무의 효율성을 비교하기 위한 여러 가지 척도들, 그리고 잉여 특성 식별에 대하여 추가적인 연구가 필요할 것으로 생각되어진다.

참 고 문 헌

[1] Aha, D.W. and R.L. Bankert, "A Comparative Evaluation of Sequential Feature Selection Algorithms," *In Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics*, Ft. Lauderdale, 1995, pp.1-7.

[2] Alumuallim, H. and T.G. Ditterich, "Learning with many Irrelevant Features," *In Proceedings of Ninth National Conference on Artificial Intelligence*, MIT Press, (1991), pp.542-547.

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

[4] Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees*, Chapman & Hall/CRC, 1998.

[5] Caruana, R. and D. Freitag, "Greedy Attribute Selection," *In Machine Learning : Proceedings of the Eleventh International Conference,* Morgan Kaufmann, (1994), pp. 28-36.

[6] Cherkauer, K.J. and J.W. Shavilik, "Growing Simpler Decision Trees to Facilitate Knowledge Discovery," *Machine Learning : In Proceedings of the second International*

Conference on Knowledge and Data Mining, AAAI press, San Mateo, (1996), pp. 315-318.

[7] Fu, Z., "A Computational Study of using Genetic Algorithms to Develop Intelligent Decision Trees," *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, (2001), pp.1382-1387.

[8] Hall, M., "Correlation-based Feature Selection of Discrete and Numeric Class Machine Learning," *In Proceedings of the International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, (2000), pp.359-366.

[9] Holmes, G. and C.G. Nevill-Manning, "Feature Selection Via the Discovery of Simple Classification Rules," *In Proceedings of th Symposium on Intelligent Data Analysis*, Baden-Baden, Germany, August, 1995.

[10] John, G.H., R. Kohavi, and P. Pfleger, "Irrelevant Features and Subset Selection Problem," *In Machine Learning : Proceedings of the Eleventh International Conference*, Morgan Kaufmann, (1994), pp.121-129.

[11] Kohavi, R. and G. John, "Wrapper for Feature Subset Selection," *In Artificial Intelligence*, Vol.97, No.1-2(1998), pp.273-324.

[12] Koller, D. and M. Sahami, "Hierarchically Classifying Documents using very Few Words," *In Machine Learning : Proceedings of the Fourteenth International Conference*, Morgan Kaufmann, (1997), pp.170-178.

[13] Kononenko, I., "Estimating Attributes : Analysis and Extension of Relief," *In Proceedings of the European Conference on Machine Learning*, (1994), pp.171-182.

[14] Kononenko, I. and E. Simec, "Induction of Decision Trees using RELIEFF," *In : Kruse, R., Viertl, R., Riccia, G. Della (eds.), CISM Lecture Notes*, Springer Verlag, (1994), pp.199-220.

[15] Koza, J.R., "Concept Formation and Decision tree Induction using the Genetic Programming Paradigm," *Parallel Problem Solving from Nature*, Berlin : Springer-Verlag, (1991), pp.124-128.

[16] Koza, J. R., *Genetic Programming*, MIT press, 1992.

[17] Lee, S. and M.Y. Huh, "A Measure of Association for Complex Data," *Computational Statistics and Data Analysis*, Vol.44, No.1-2(2003), pp.211-222.

[18] Murthy, S.K., "Automatic Construction of Decision Trees from Data : A Multidisciplinary Survey," *In Data Mining and Knowledge Discovery*, No.2(1998), pp.345-389.

[19] Papagelis, A. and D. Kalles, "Breeding Decision Trees using Evolutionary Techniques," *ICML*, (2001), pp.393-400.

[20] Pfahringer, B., "Compression-based Feature Subset Selection," *In Proceedings of the IHCAI-95 Workshop on Data Engineering for Inductive Learning*, (1995), pp.109-119.

[21] Quinlan, J.R., *C4.5 : Programs for Machine Learning*, San Mateo, CA : Morgan Kaufmann, 1993.

[22] Setiono, R. and H. Liu, "Chi2 : Feature Selection and Discretization of Numeric Attributes," *In Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence*, (1995), pp.388-391.

[23] Soule, T., "Code Growth in Genetic Programming," *PhD thesis*, University of Idaho, Moscow, Idaho, USA, 1998.

[24] Vafail, H. and K. De Jong, "Genetic Algorithms as a Tool for Restructuring Feature Space Representations," *In Proceedings of the International Conference on Tools With A. I.*, IEEE Computer Society Press, 1995.

[25] Witten, I.H. and F. Eibe, *Data Mining*, Morgan and Kaufmann, 1990.

[26] http://www.cs.ucl.ac.uk/external/A.Qureshi/gpsys_doc.html.

[27] http://www.cs.waikato.ac.nz/ml/weka.

[28] http://www.ics.uci.edu/~mlean/MLRepository.html.

[29] http://www.r-project.org.