응용통계연구 제 14권 2호, 2001년, pp. 475-486

데이터마이닝 패키지에서 변수선택 편의에 관한 연구 *

송문섭[1] 윤영주[2]

요 약

데이터마이닝 패키지에 구현된 분류나무 알고리즘 가운데 CART, CHAID, QUEST, C4.5에서 변수선택법을 비교하였다. CART의 전체탐색법이 편의를 갖는다는 사실은 잘 알려졌으며, 여기서는 상품화된 패키지들에서 이들 알고리즘의 편의와 선택력을 모의실험 연구를 통하여 비교하였다. 상용 패키지로는 CART, Enterprise Miner, AnswerTree, Clementine을 사용하였다. 본 논문의 제한된 모의실험 연구 결과에 의하면 C4.5와 CART는 모두 변수선택에서 심각한 편의를 갖고 있으며, CHAID와 QUEST는 비교적 안정된 결과를 보여주고 있었다.

주요용어: 전체탐색법, 변수선택 편의, CART, CHAID, C4.5, QUEST

1. 서 론

데이터마이닝을 위한 통계적 도구로서 분류나무(classification tree)는 매우 유용하고 널리 사용되는 기법이다. 분류나무는 예측변수(predictor variable)를 이용하여 목표변수(target variable)를 예측하는 규칙의 집합으로서, 목표변수의 수준이 알려진 훈련용 데이터를 반복적으로 분할하여 분류나무를 만든다. 이 때 각 분할 과정으로 나무의 마디(node)가 만들어지며, 각 마디에서는 데이터를 분할하는 방법인 분리기준(splitting rule)을 결정한다. 이와 같은 분리기준은 분류나무 알고리즘에서 가장 중요한 역할을 하게된다.

지금까지 많은 분류나무 알고리즘이 제안되었으며, 그 가운데 잘 알려진 알고리즘으로는 CHAID(Kass, 1980), CART(Breiman, Friedman, Olshen and Stone, 1984), C4.5(Quinlan, 1993), QUEST(Loh and Shih, 1997) 등이 있다. 최근에 Kim and Loh(1999)는 CRUISE라는 새로운 분류나무 알고리즘을 제안했지만 아직 상용화된 것은 아니므로 본 연구에서는 제외하였다. 이와 같은 방법들은 E-miner(Enterprise Miner), CART, S-Plus, AnswerTree, Clementine 등과 같은 상용 소프트웨어 패키지에 구현되어 있으므로, 이들 패키지를 직접 비교하기로 한다. 많은 알고리즘들은 패키지에 구현될 때 악간의 수정을 하게 되므로, 본 연구에서는 패키지에서 가능하면 디폴트 옵션(default option)을 사용하여 실행하기로 한다.

* 본 연구는 Brain Korea 21 Project의 지원에 의한 것임

1) (151-747) 서울시 관악구 신림동 산 56-1, 서울대학교 통계학과, 교수
E-mail: songms@plaza.snu.ac.kr
2) (151-747) 서울시 관악구 신림동 산 56-1, 서울대학교 통계학과, 박사과정
E-mail: youngjoo@stats.snu.ac.kr

한 변수에 의한 분리에서 분리기준은 분리변수의 선택과 선택된 변수에 의한 분리점 또는 분리집합으로 특징지을 수 있다. 한편 CART에서 사용하는 전체탐색법(exhaustive search method)은 편의가 심각한 것으로 알려져 있다. 즉, 범주의 개수가 많은 범주형 예측변수가 있을 경우에 이 변수가 목표변수와 연관성이 없어도 분리변수로 선택될 가능성이 커지는 문제점이 있다. 이 문제에 대한 부분적인 연구는 되어 있으나, 본 논문에서는 CHAID와 C4.5(또는 C5.0)를 포함하여 비교하고자 하며, 이들이 구현된 패키지에서 적접 비교하려고 한다.

소프트웨어 패키지의 수행능력을 비교하기 위하여 몬테칼로 모의실험을 시행하였다. 즉, 모의실험 자료를 컴퓨터로 생성하고, 이들에 데이터마이닝 패키지를 적용시켰으며, 변수선택 확률을 추정하여 편의와 선택력을 조사하였다. 모의실험에 의하면 C4.5와 CART에 기초한 패키지에서는 모두 변수선택에서 심각한 편의가 있는 것으로 나타났으며, CHAID는 비교적 편의가 덜 심각하고, QUEST가 가장 안정된 결과를 보여주고 있었다. 그러나 QUEST는 연속형 변수에서 정규성의 가정에 예민하므로, 변수 선택력이 CART나 CHAID보다 훨씬 떨어질 수 있음을 보여주었다.

2. 변수선택 알고리즘

이 절에서는 분류나무를 생성할 때 사용하는 변수선택 알고리즘에 대하여 간단히 요약하고, 그 특징을 설명하도록 한다.

CHAID(Chi-squared Automatic Interaction Detection, Kass, 1980)는 최적의 분리기준을 찾기 위해 카이제곱 통계량을 이용한다. 즉, 연속형 예측변수는 구간별로 그룹화 하여 범주형으로 변환시킨 후에, 각 예측변수와 목표변수의 분할표로부터 카이제곱 통계량을 계산한다. 모든 예측변수에 대하여 카이제곱 통계량을 구하고, 이 가운데 가장 유의한 변수를 선택하여 분리변수로 사용하게 된다.

CHAID에서는 분리변수의 각 범주가 하나의 부마디(sub-node)를 형성하는 다원분리(multiway split)를 실행한다. 다원분리와 이진분리(binary split)는 각각의 장단점이 있으나, 이것이 분리변수 선택에 영향을 주는 것은 아니다. CHAID 알고리즘에서 변수선택 부분을 설명하면 다음과 같다 (Kass, 1980).

1. 각 예측변수에 대해, 예측변수와 목표변수의 범주를 수준으로 하는 분할표를 만들고 다음의 2단계와 3단계를 시행한다.

2. 예측변수 범주의 각 쌍과 목표변수의 범주를 사용한 부분할표를 만들고, 가장 유의성이 약한 쌍의 유의확률이 주어진 임계값보다 크면, 이들 두 쌍을 하나의 범주로 병합하는 과정을 반복한다.

3. 3개 이상의 범주가 병합된 새로운 범주에 대하여는, 가장 유의성이 있는 이진분류를 찾아서 유의확률이 임계값보다 작으면 분리를 시행하고 2단계로 돌아간다.

4. 새로 얻은 각 예측변수를 사용하여, 각 예측변수와 목표변수의 분할표에서 유의성을 구하고, 가장 작은 유의확률이 임계값보다 작으면 대응되는 예측변수의 범주에 따라

관측값을 분할한다. 다만, 병합과정에서 여러 단계의 검정을 거치므로 Bonferroni 수정된 유의확률을 사용한다.

CART(Classification and Regression Tree, Breiman et al., 1984)의 전체탐색법은 최선의 이진분리를 찾는다. 즉 각 마디에서, 모든 가능한 분리에 대하여 불순도 측도(impurity measure)를 계산하고, 불순도 측도를 최소로 하는 분리기준을 선택한다. 불순도 측도로는 지니 지수(Gini index)를 많이 사용한다. 알고리즘은 다음과 같다.

1. 각 순서형 예측변수에 대하여는, 모든 '$X \le c$' (c는 연속된 두 데이터의 중앙값) 형태의 분리기준에 대한 불순도를 계산한다.
2. 각 범주형 예측변수에 대하여는, 모든 '$X \in A$' (A는 X 범주의 부분집합) 형태의 분리기준에 대한 불순도를 계산한다.
3. 불순도 측도를 최소로 하는 분리기준을 선택한다.

CART의 전체탐색법에서 n개의 값을 갖는 순서형 예측변수에는 모두 $n-1$개의 가능한 분리기준이 있고, k개의 범주를 갖는 범주형 예측변수에는 $2^{k-1}-1$개의 가능한 분리기준이 있다. 따라서 한 예측변수는 연속형이고 다른 예측변수는 범주의 개수가 적은 범주형이면 연속형에 대하여는 더 많은 횟수로 불순도 계산을 하므로, 두 변수가 같은 조건이라면 연속형 변수가 선택될 기회가 많아진다. 더욱 심각한 것은 범주형 변수에서 범주의 개수가 클 때이다. 즉, 범주의 개수가 증가함에 따라 모든 가능한 분리기준의 개수는 지수적으로 증가하므로 범주형 변수가 선택될 기회가 더 많아지게 된다. 예를 들어 $k=15$일 때 가능한 분리기준의 개수는 $2^{15-1}-1=16,383$개가 있다. 이와 같은 이유로 CART의 전체탐색법은 범주의 개수가 많은 예측변수 방향으로 심각한 편의가 존재한다.

분류 알고리즘 C4.5는 Quinlan(1993)이 제안했으며, Clementine 등에 구현되어 많이 사용하고 있는 알고리즘이다. C4.5의 특징은 순서형 예측변수에 대하여는 이진분리를 시행하고, 범주형 예측변수에 대하여는 다원분리를 시행하는 것이다. C4.5에서 사용하는 분리 측도는 이득비율(gain ratio)이다. 먼저 이득(gain)은 엔트로피(entropy)의 감소로 정의되며, 이득에 기초한 기준은 범주의 개수가 많은 변수로의 편의가 심각하다. 예를 들어 ID변수는 각 개인을 하나의 부마디로 분리시킬 때 이득이 최대가 된다. 그러나 이러한 분리는 예측면에서는 의미가 없다. 따라서 Quinlan(1993)은 '(엔트로피의 감소)/(분리의 엔트로피)'로 정의되는 이득비율을 분리측도로 제안하였다. C4.5에서는 이와 같이 표준화된 이득의 개념을 사용하여 분리 측도로 사용하지만, 그럼에도 불구하고 범주의 개수가 많은 예측변수 방향으로의 편의는 매우 심각할 것으로 예상된다.

C5.0은 C4.5를 발전시킨 것이다. 예를 들어 C5.0에서는 범주형 변수의 분리에서 범주의 병합 및 재분리가 디폴트로 주어진다. 본 논문에서는 C5.0을 사용하기로 한다. C5.0에서 변수선택 부분은 다음과 같다.

1. 각 순서형 예측변수에 대해서는, 모든 '$X \le c$' (c는 연속된 두 데이터의 중앙값) 형태의 분리기준에 대한 이득비율을 계산한다.

2. 각 범주형 예측변수에 대해서는, 이득비율을 가장 많이 개선시키는 범주의 쌍을 병합한다. 이득비율이 개선되지 않을 때까지 이 과정을 반복한다.

3. 이득비율을 가장 많이 향상시키는 분리기준을 선택한다.

QUEST(Quick, Unbiased, Efficient, Statistical Tree, Loh and Shih, 1997)는 변수선택 편의를 줄이기 위해 변수선택의 단계와 선택된 변수에 기초한 분리기준을 찾는 단계를 나눈 것이 특징이다. 변수선택은 ANOVA F-검정이나 분할표의 카이제곱검정에서 p-값을 계산하고 가장 작은 p-값에 대응되는 변수를 분리변수로 선택한다. QUEST에서 변수선택 알고리즘을 간단히 요약하면 다음과 같다.

1. 순서형 예측변수에 대해서는, ANOVA F-검정의 p-값을 계산한다.

2. 범주형 예측변수에 대해서는, 예측변수와 목표변수의 분할표에서 카이제곱검정의 p-값을 계산한다.

3. 1, 2단계에서 가장 작은 p-값이 Bonferroni 수정된 임계값보다 작으면 그에 대응되는 변수를 분리변수로 선택한다.

4. 그렇지 않으면, 순서형 예측변수에 대하여 Levene F-검정의 p-값을 계산하고 Bonferroni수정된 임계값과 비교한다. p-값이 임계값보다 작으면 대응되는 변수를 선택하고, 아니면 1, 2단계에서 구한 가장 작은 p-값에 대응되는 변수를 분리변수로 선택한다.

이와 같이 QUEST에서는 목표변수에 대한 예측변수의 기여도를 F-검정 또는 카이제곱검정의 p-값으로 측정함으로서, 변수선택의 편의를 줄이도록 하였다.

상용 패키지에서 변수선택 알고리즘을 비교하기 위하여 E-miner (버젼 3.0, SAS Institute Inc.), CART (버젼 3.6.3, Salford Systems), Clementine (버젼 5.2.1, SPSS Inc.), AnswerTree (버젼 2.1, SPSS Inc.)를 사용하였다. E-miner에서는 디폴트 옵션인 'Chi-square test' 옵션을 사용하고, CART에서는 디폴트 옵션인 'Gini' 옵션을 사용했다. Clementine에서는 'Build C5.0'을 선택하고, AnswerTree에서는 CHAID와 QUEST 알고리즘을 선택했다. 패키지와 알고리즘의 관계를 표 2.1에 요약하였다.

표 2.1: 소프트웨어 패키지와 알고리즘

기 호	패키지	기본 알고리즘
CART	CART	전체탐색(지니 지수)
E-χ^2	E-miner	CHAID
CHAID	AnswerTree	CHAID
QUEST	AnswerTree	QUEST
C5.0	Clementine	C5.0

E-miner에서 지니 지수 옵션을 사용한 결과와 AnswerTree에서 C&RT 옵션을 사용한 결과는 모두 CART의 결과와 매우 비슷하다. S-Plus는 지니 지수 대신에 엔트로피(또는 이탈도, deviance)를 사용하지만 목표변수의 개수가 2개인 문제에서는 지니 지수와 엔트로피의 차이가 거의 없으므로 S-Plus의 결과는 CART와 매우 유사하다. 따라서 CART의 결과와 유사한 E-miner의 지니 지수 옵션, AnswerTree의 C&RT 옵션 및 S-Plus는 본 논문에 포함시키지 않았다.

3. 모의실험 설계와 결과 비교

3.1. 변수선택 편의를 비교하기 위한 설계

변수선택 편의와 선택력을 비교하기 위하여 몬테칼로 모의실험 연구를 수행하였다. 먼저 편의를 조사하기 위하여 상호독립인 5개의 예측변수와 이들과 독립인 목표변수를 선택한다(이들을 '독립모형'이라 부르기로 한다). 목표변수는 $\{1,2\}$의 값을 같은 확률로 갖는 이진변수이며, 변수 X_1과 X_2는 각각 정규분포와 지수분포에 따르는 연속형 변수이다. 변수 X_3는 $\{1,2,3,4\}$의 값에서 균등분포에 따르는 순서형 변수이며, X_4는 $\{1,2\}$의 값을 같은 확률로 갖는 범주형 변수이다. X_5는 범주형으로서 $\{1,2,\cdots,k\}$의 값을 같은 확률로 갖는다. 따라서 k는 X_5의 범주의 개수를 나타내며, 모의실험에서는 $k=5$와 $k=15$의 두 가지 경우를 고려하였다.

관측값의 개수인 N은 200, 500, 1000의 세 가지 경우를 다루었다. 5개의 예측변수와 목표변수가 상호독립인 경우에 각 X_i가 분리변수로 선택될 확률은 0.2이어야 한다. 그러나 분리변수의 선택법에 따라 각 변수의 선택확률이 달라지므로, 이들 선택확률을 몬테칼로 모의실험을 반복시행하여 추정한다. 즉, 몬테칼로 반복수를 300으로 하고, 300회 가운데 각 X_i가 선택된 횟수로 선택확률을 추정한다. 따라서 각 확률 추정값의 표준오차는 0.029이다.

각 알고리즘의 선택력을 알아보기 위해서는 '이동모형'(shifted model)에서 모의실험을 시행하였다. 즉, 정규확률변수인 X_1을 다음과 같은 분포에서 생성하였다.

$$X_1 \sim \begin{cases} N(0.2,1), & \text{범주 1일 때} \\ N(0.0,1), & \text{범주 2일 때} \end{cases}$$

그 밖의 다른 변수들은 독립모형과 같다. 따라서 이동모형으로 생성된 경우에는 X_1이 선택될 확률이 클수록 선택력이 높은 (따라서 우수한) 변수선택 알고리즘이라고 할 수 있다.

데이터는 SAS에서 normal, ranexp, ranuni 함수를 이용하여 생성하였으며, 생성된 데이터 각각에 데이터마이닝 패키지를 적용시켰다. 각 변수가 선택될 확률은 뿌리 마디(root node)에서 그 변수가 선택되는 횟수로 추정하였다. 결과는 표 3.1에 요약되어 있다.

3.2. 실험 결과

CART에서는 예상했던 대로 연속형 변수나 범주의 개수가 많은 범주형 변수로의 편의

표 3.1: 변수선택 확률의 추정값

알고리즘	독립모형 x_1	x_2	x_3	x_4	x_5	이동모형 x_1	x_2	x_3	x_4	x_5
				$k=5$,	$N=200$					
CART	.383	.350	.060	.013	.193	.597	.243	.013	.017	.130
E-χ^2	.070	.123	.277	.293	.237	.320	.117	.150	.207	.207
CHAID	.127	.117	.250	.350	.157	.393	.100	.143	.210	.153
QUEST	.203	.173	.223	.190	.210	.500	.120	.110	.107	.163
C5.0	.017	.000	.107	.137	.740	.097	.023	.097	.107	.677
				$k=5$,	$N=500$					
CART	.387	.383	.033	.030	.167	.793	.127	.020	.007	.053
E-χ^2	.130	.190	.193	.270	.217	.563	.067	.100	.183	.087
CHAID	.147	.193	.237	.290	.133	.613	.047	.137	.157	.047
QUEST	.187	.233	.210	.183	.187	.773	.030	.097	.050	.050
C5.0	.003	.007	.093	.153	.743	.163	.007	.103	.133	.593
				$k=5$,	$N=1,000$					
CART	.360	.460	.050	.003	.127	.907	.080	.000	.007	.007
E-χ^2	.117	.190	.233	.270	.190	.750	.053	.053	.077	.067
CHAID	.133	.160	.243	.330	.133	.837	.027	.040	.080	.017
QUEST	.177	.270	.233	.153	.167	.933	.020	.007	.030	.010
C5.0	.000	.000	.153	.107	.740	.347	.013	.087	.100	.453
				$k=15$,	$N=200$					
CART	.103	.123	.017	.003	.753	.213	.077	.007	.013	.690
E-χ^2	.133	.167	.223	.370	.107	.303	.153	.177	.297	.070
CHAID	.163	.203	.237	.380	.017	.360	.150	.200	.280	.010
QUEST	.237	.213	.213	.200	.137	.457	.153	.167	.133	.090
C5.0	.003	.007	.007	.000	.983	.040	.003	.017	.013	.927
				$k=15$,	$N=500$					
CART	.077	.137	.007	.003	.777	.480	.050	.010	.003	.457
E-χ^2	.120	.210	.250	.340	.080	.590	.087	.103	.143	.077
CHAID	.163	.193	.247	.387	.010	.630	.080	.107	.180	.003
QUEST	.190	.207	.233	.170	.200	.733	.087	.053	.050	.077
C5.0	.000	.000	.003	.003	.993	.060	.000	.003	.010	.927
				$k=15$,	$N=1,000$					
CART	.133	.093	.023	.000	.750	.750	.020	.000	.000	.230
E-χ^2	.147	.157	.267	.350	.080	.847	.033	.043	.060	.017
CHAID	.160	.177	.283	.377	.003	.877	.033	.047	.043	.000
QUEST	.217	.220	.257	.147	.160	.943	.020	.020	.003	.013
C5.0	.000	.000	.027	.003	.970	.177	.000	.020	.007	.797

가 심각한 수준이다. 독립모형에서 $k=5$일 때 연속형인 X_1, X_2가 선택될 확률이 범주형보다 훨씬 큼을 알 수 있다. 예를 들어 $k=5$, $N=200$일 때 $\{X_1, X_2\}$가 선택될 확률은 0.74로 추정된 반면에 $\{X_3, X_4, X_5\}$가 선택될 확률은 0.26으로 추정된다. $k=15$로서 X_5의 범주의 개수가 많은 경우에는 X_5가 선택될 확률이 다른 변수보다 압도적으로 높은 것으로 나타났다. 이동모형에서 $k=5$일 때는 CART의 선택력이 우수한 것으로 나타났다. X_1으로 약간의 편의가 존재함을 감안하더라도 QUEST와 함께 가장 높인 선택력을 보인다. $k=15$일 때는 X_5로의 편의가 있기 때문에 X_1의 선택력이 떨어지나, 전체적으로는 다른 알고리즘과 비교하여 경쟁력이 있음을 보여준다.

E-miner에서 CHAID 알고리즘인 'Chi-square test' 옵션을 사용한 E-χ^2는 비교적 안정된 결과를 보여주고 있다. 먼저 독립모형에서 X_3, X_4로의 편의가 약간 존재하지만 CART의 전체탐색법 만큼 심각하지는 않다. 그러나 $k=15$일 때는 X_5의 선택확률이 감소하는 대신에 X_4의 선택확률이 증가하는 결과를 볼 수 있다. 이동모형에서 X_1의 선택력은 중간인 것으로 보인다.

AnswerTree에서 CHAID는 E-χ^2와 기본적으로 같은 알고리즘이므로 결과도 매우 비슷하다. 독립모형에서 X_4로의 편의가 E-χ^2보다는 약간 크게 보이지만 심각한 차이는 아니다. 이동모형에서 CHAID가 E-χ^2보다 X_1의 선택력이 약간 우세하지만 이들의 차이도 심각한 것은 아니다. AnswerTree에서 QUEST는 편의와 선택력 모두에서 우수한 것으로 나타났다. 독립모형에서 심각한 편의는 없으며, 이동모형에서 선택력도 우수한 결과를 보이고 있다. 즉, $k=5$일 때는 CART와 비슷한 선택력을 보인 반면에 $k=15$일 때는 제일 높은 선택력을 보이고 있다. 따라서 연속형 변수가 정규분포 및 지수분포에 따르는 표 3.1의 모형에서는 QUEST가 변수선택의 측면에서 좋은 알고리즘인 것으로 생각된다.

Clementine의 C5.0은 편의가 CART보다 훨씬 심각하게 나타난다. $k=15$일 때는 X_5로의 편의가 너무 심각해서 이동모형에서도 선택력이 심각하게 떨어지는 결과를 보이고 있다. C5.0은 실제로 우수한 알고리즘으로 인정되어 널리 사용되고 있는 알고리즘이지만, 변수선택의 측면에서는 범주의 개수가 많은 범주형 변수로의 편의가 심각한 것으로 보인다.

표 3.1에서 관측값의 개수인 N의 변화에 따른 선택확률의 차이는 크지 않은 것으로 보인다. 다만, 이동모형에서 N이 증가함에 따라 X_1의 선택력도 약간 증가함을 알 수 있다.

3.3. 변수 선택력의 비교

표 3.1에서는 변수의 형태에 따른 변수선택 편의를 알아보고, 이동모형에서 이들 편의가 어떻게 극복되는지를 알아보는 모의실험이었다. 그러나 각 알고리즘에서 변수 선택력만 보려면 X_1부터 X_5를 모두 같은 성질을 갖는 변수로 하고, X_1을 이동모형으로 만들어 각 알고리즘에서 X_1의 선택력을 알아볼 필요가 있다. 또한 QUEST에서는 F-검정을 시행하므로 자료가 정규분포일 때는 다른 알고리즘에 비해 유리한 조건을 가질 수 있다. 따라서 각 변수를 모두 오염정규분포에서 생성하는 모의실험을 수행하기로 한다. 즉, 80%의 데이터는 $N(0,1)$에서 생성하고 20%의 데이터는 $N(0,5^2)$에서 생성하였으며, 표 3.1에서와 마찬가지로 X_1에 대하여는 목표변수의 값이 1일 때 평균이 0.2만큼 이동되는 이동모형을 사용하였다. 관측값의 개수인 N은 1,000인 경우만 생각했으며, 결과를 표 3.2에 요약하였다.

표 3.2: 오염정규분포에서 변수선택 확률의 추정값

알고리즘	x_1	x_2	x_3	x_4	x_5
CART	.653	.107	.083	.067	.090
E-χ^2	.647	.110	.083	.060	.100
CHAID	.713	.070	.070	.060	.087
QUEST	.493	.150	.097	.090	.170

표 3.2의 모형에서도 표 3.1의 이동모형에서와 마찬가지로 X_1이 선택될 확률이 클수록 변수 선택력이 우수한 알고리즘이라고 할 수 있다. 그러나 C5.0을 구현한 Clementine에서는 약 72%에서 분리가 일어나지 않았으며, 분리가 일어나도록 조정하는 옵션이 없었기 때문에 다른 알고리즘과의 비교가 어려워 표에서 제외하였다. 표 3.1의 독립모형에서는 모든 경우에 분리가 일어났음에도 불고하고 표 3.2의 모형에서는 분리가 일어나지 않은 것은 C5.0의 변수선택 편의가 그만큼 심각한 때문인 것으로 생각된다.

표 3.1의 이동모형에서는 QUEST가 변수 X_1 선택력이 가장 우수한 것으로 나타났으나. 표 3.2에서는 QUEST의 변수 선택력이 매우 낮은 것으로 나타났다. 이는 자료들이 정규성의 조건을 만족시키지 못하므로 F-검정의 효율이 떨어지기 때문인 것으로 판단된다. 즉 QUEST는 정규성의 가정에 예민하며, 대부분의 데이터마이닝 자료들이 정규성의 조건을 만족시키지 못함을 감안할 때 QUEST의 변수 선택력은 상당히 떨어지는 것으로 생각된다. CHAID는 가장 높은 선택력을 보이고 있으며, CART와 E-χ^2는 결과가 매우 비슷하면서 CHAID 다음으로 높은 선택력을 보이고 있다. 표 3.1와 표 3.2의 제한된 모형의 결과로 각 알고리즘의 변수 선택력을 평가하기는 어렵지만, QUEST는 정규성의 가정에 예민하며 나머지 CART, E-χ^2, CHAID는 모두 자료의 분포에 로버스트한 방법인 것으로 판단된다.

3.4. 가지치기의 영향

Breiman et al.(1984)이 제안한 CART 알고리즘에서는 나무를 충분히 키운 다음에 가지치기(pruning)를 통하여 최적인 나무를 선택한다. 그러나 표 3.1와 표 3.2의 모의실험에서는 각 변수의 선택확률을 추정하여 편의와 선택력을 알아보는 것이 목적이므로, 뿌리 마디에서 언제나 분리가 일어나도록 옵션을 선택하였다. 즉, CART에서는 *Model Setup* 대화창의 *Testing* 탭에서 'no independent testing' 옵션을 선택하여 가지치기를 막았으며, E-miner의 E-χ^2에서는 *Tree* 노드의 *Basic* 창에서 *Chi-square test*의 'significance level'을 1.0으로 조정하고, 'Maximum depth of tree'를 2로 조정하여 뿌리 마디에서 정지규칙이 적용되지 않고 분리가 최소한 한번은 일어나도록 시행하였다.

그러나 독립모형의 경우에 가지치기 또는 정지규칙에 의해 뿌리 마디에서 분리가 일어나지 않는다면 변수선택의 편의는 문제가 되지 않는다. Kim and Loh(1999)는 이를 확인하기 위하여 독립모형에서 가지치기를 허용하는 모의실험을 시행하였다. 그들의 결과에 의

표 3.3: 독립모형에서 분리가 일어날 확률과 조건부 변수선택 확률의 추정값

알고리즘	k	N	분리확률	조건부 변수선택 확률				
				x_1	x_2	x_3	x_4	x_5
CART	5	200	.407	.361	.287	.057	.016	.279
		500	.533	.313	.394	.050	.031	.213
		1,000	.603	.331	.436	.066	.006	.160
	15	200	.477	.056	.091	.007	.000	.846
		500	.547	.079	.098	.006	.000	.817
		1,000	.503	.146	.066	.026	.000	.762
E-χ^2	5	200	.520	.077	.122	.276	.308	.218
		500	.587	.165	.227	.188	.250	.170
		1,000	.497	.154	.195	.282	.228	.141
	15	200	.463	.180	.230	.281	.259	.050
		500	.470	.121	.284	.291	.262	.043
		1,000	.493	.223	.182	.277	.284	.034

하면, CART는 본 논문의 표 3.1와 비슷한 독립모형에서 약 40% 정도는 적어도 한번 이상의 분리가 일어난 것으로 보고되었다.

이제 CART와 E-χ^2의 각각에서 가지치기와 정지규칙을 적용시킬 경우에 표 3.1의 독립모형에서 분리가 일어날 확률과, 분리가 일어났을 때 각 변수가 선택될 확률을 알아보자. 이를 위해 CART에서는 *Model Setup* 창의 *Testing* 탭에서 디폴트 옵션인 '10-fold cross validation'을 선택하고, E-χ^2에서는 *Chi-square test*의 'significance level'을 디폴트 값인 0.20으로 선택한다. 이와 같이 디폴트 값으로 조정된 CART와 E-χ^2를 표 3.1의 독립모형 자료 300개에 적용시켰으며, 결과는 표 3.3에 요약하였다. 표에서 '분리확률'은 최소한 한번의 분리가 일어날 확률의 추정값이며, '조건부 변수선택 확률'은 분리가 일어났을 때 각 변수가 분리변수로 선택될 확률의 추정값이다. AnswerTree의 CHAID는 E-miner의 E-χ^2와 기본적으로 같은 알고리즘이지만 실제 구현에서는 약간의 차이가 있다. 예를 들어 CHAID에서는 카이제곱 검정의 유의수준을 디폴트 값인 0.05로 설정했기 때문에 전체의 약 80% 정도에서 분리가 일어나지 않았다. 한편 QUEST에서는 *Advanced Options*에서 가지치기를 위한 옵션조정이 가능하며, 디폴트 값으로 *Grow and Prune*을 시행할 경우에 대부분 분리가 일어난다. 따라서 CHAID와 QUEST는 CART 및 E-χ^2와 비교가 어렵다고 판단되어 표에서 생략했다.

표 3.3의 결과에 의하면 전체의 약 40% ~ 60%의 경우에 최소한 한번의 분리가 일어남을 알 수 있다. CART와 E-χ^2에서 분리가 일어나는 확률은 비슷한 수준이다. 한편, 분리가

일어난 경우에 CART와 E-χ^2의 조건부 변수선택 확률은 표 3.1의 결과와 매우 비슷하다. 즉, 범주형 변수인 X_5에서 범주의 개수가 $k = 15$로서 많은 경우에는 CART의 변수선택 편의가 매우 심각함을 알 수 있다. 그러나 E-χ^2는 표 3.1에서와 같이 편의의 측면에서는 비교적 안정된 결과를 보여주고 있다.

4. 결 론

본 논문의 제한된 모의실험 연구 결과에 의하면 한 알고리즘이 다른 알고리즘보다 항상 우수한 경우는 없었다. 그러나 일부 알고리즘에서는 변수선택 편의가 심각한 반면에 선택력에서는 우수한 결과를 보이기도 했다.

CHAID 알고리즘을 사용하는 E-miner의 E-χ^2와 AnswerTree의 CHAID는 편의와 선택력에서 로버스트한 결과를 보여주었다. 즉, 변수선택 편의는 심각하지 않지만, 이동모형에서 변수 선택력이 가장 우수한 것은 아니다. 그러나, 표 3.2의 오염정규분포에서는 변수 선택력이 가장 우수한 것으로 나타났다.

CART는 변수선택 편의는 심각한 수준이지만 변수 선택력은 우수한 것으로 나타났다. 즉, 범주의 개수가 많은 범주형 변수로의 편의가 CHAID나 QUEST에 비하면 심각하게 큰 반면에 오염정규분포에서는 CHAID와 같이 높은 변수 선택력을 보이고 있다.

QUEST는 편의의 측면에서는 가장 우수하지만, 변수선택 과정에서 F-검정을 시행하므로 정규성의 가정에 예민한 단점이 있는 것으로 판단된다.

C4.5에 기초한 Clementine은 변수선택 편의가 CART보다 더 심각하며, 따라서 이런 편의가 존재하는 경우에는 변수 선택력도 떨어짐을 확인할 수 있었다.

참고문헌

[1] Breiman, L., Friedman, J., Olshen, R. and Stone, C (1984). *Classification and Regression Trees*, Chapman and Hall, New York, NY.

[2] Kass, G.V. (1980). An exploratory technique for investigating large quantiles of categorical data, *Applied Statistics*, **29**, 119-127.

[3] Kim, H. and Loh, W.-Y. (1999). Classification trees with unbiased multiway splits, Technical Report 1012, Department of Statistics, University of Wisconsin-Madison.

[4] Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees, *Statistica Sinica*, **7**, 815-840.

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

[6] Salford Systems (1997). *CART*, Salford Systems, San Diago, CA.

[7] SAS Institute Inc. (1998). *Decision tree and Regression Node Version 2.0 Online Documentation*, SAS Institute Inc.

[8] SPSS Inc. (1998). *AnswerTree 2.0 User's Guide* , SPSS Inc, Chicago IL.

[2001년 3월 접수, 2001년 9월 채택]

A Study on Variable Selection Bias in Data Mining Software Packages*

Moon Sup SONG[1] Young Joo YOON [2]

ABSTRACT

We compare the variable selection methods in classification tree algorithms such as CART, CHAID, QUEST, and C4.5. It is well known that the exhaustive search method of CART has serious bias problem in variable selection toward many-valued categorical predictors. In this paper we compare the commercial softwares in terms of variable selection bias and power. A Monte Carlo simulation study was performed to compare the softwares such as CART, Enterprise Miner, AnswerTree, and Clementine. The results show that the softwares based on C4.5 and the exhaustive search of CART are seriously biased in variable selection. The bias of CHAID is less serious than that of CART or C4.5. QUEST does not show any serious bias. But, the unbiased methods are not necessarily most powerful in variable selection.

Keywords: Exhaustive search method; Bias in variable selection; CART; QUEST.

* This research was supported in part by the Brain Korea 21 Project.

1) Professor, Department of Statistics, Seoul National University.
E-mail: songms@plaza.snu.ac.kr

2) Graduate Student, Department of Statistics, Seoul National University.
E-mail: youngjoo@stats.snu.ac.kr