Proceedings of Joint Conference of The Korean Data Analysis Society and Korean Data And Information Science Society, October 17-18, 2003, pp. 215-228

Exploration of CHAID Algorithm by Sampling Proportion

Hee-Chang Park[1], *Kwang-Hyun Cho*[2]

Abstract

Decision tree algorithms are used extensively for data mining in many domains such as retail target marketing, fraud dection, data reduction and variable screening, interaction effect identification, category merging and discretizing continuous variable, etc. CHAID(Chi-square Automatic Interaction Detector), is an exploratory method used to study the relationship between a dependent variable and a series of predictor variables. CHAID modeling selects a set of predictors and their interactions that optimally predict the dependent measure. In this paper we explore CHAID algorithm in view of accuracy and speed by sampling proportion.

Keywords : data mining, decision trees, CHAID, random samplimg

1. 서론

데이터 마이닝에서 사용하는 기법에는 연관성규칙(association rule), 의사결정나무(decision tree), 신경망 분석(neural network analysis), 클러스터링(clustering), 유전자 알고리즘(geneic algorithm), 베이지안 네트워크(bayesian network), 메모리-기반 추론(memory-based reasoning) 등이 있다. 이 중에서 의사결정나무는 의사결정 규칙(decision rule)을 도표화하여 관심대상이 되는 집단을 몇 개의 소집단으로 분류하거나 예측을 수행하는 분석방법으로 분류와 예측을 목적으로 하는 신경망 분석, 판별분석(discriminant analysis), 회귀분석(regression analysis)등의 다른 분석들에 비해 연구자가 분석과정을 쉽게 이해하고 설명할 수 있다는 장점이 있다.

의사결정나무 알고리즘은 분류 또는 예측을 목적으로 하는 어떤 경우에도 사용될 수 있으나, 분석의 정확도보다는 분석과정의 설명이 필요한 경우에 더 유용하게 사용된다. 특히 시장세분화, 고객세분화 등의 세분화(segmentation), 고객을 신용도에 따라 우량/불량으로 분류하는 분류 문제, 고객속성에 따라서 대출한도액을 예측하는 예측, 차원축소 및 변수선택(data reduction and variable

[1]Professor, Department of Statistics, Changwon National University, Changwon, Kyungnam, 641-773, Korea. E-mail : hcpark@sarim.changwon.ac.kr

[2]Graduate Student, Department of Statistics, Changwon National University, Changwon, Kyungnam, 641-773, Korea

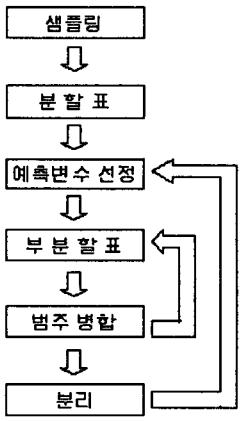
screening), 교호작용효과의 파악(interaction effect identification), 범주의 병합(category merging) 또는 연속형 변수의 이산화(discretizing continuous variable) 등의 분야에서 유용하게 활용되고 있다. 의사결정나무기법은 탐색(exploration)과 모형화(modeling)라는 두 가지 특성을 모두 가지고 있다고 할 수 있다.

현재까지 연구를 살펴보면 의사결정나무분석을 수행하기 의한 다양한 분리기준, 정지규칙, 가지치기 방법들이 제안되어 있으며, 이들을 어떻게 결합하느냐에 따라서 서로 다른 의사결정나무 형성방법이 만들어진다. 또한 정확하고 빠르게 의사결정나무를 형성하기 위해서 다양한 알고리즘이 제안되어 있고, 보다 개선된 알고리즘들이 계속 연구되어 발표되고 있다. 의사결정나무 알고리즘에는 Hartigan(1975)이 제안한 CHAID(Chi-squared Automatic Interaction detecion), Breiman(1984)이 제안한 CART(Classification and Regression Trees), Quinlan(1993)에 의해 제안된 C4.5, 그리고 Lon와 Shin(1997)이 제안한 QUEST(Quick, Unbiased, Efficient, Statistical Tree) 알고리즘 등이 있으며, 이들은 많은 소프트웨어 회사에 의해서 다양한 제품으로 상용화되고 있다. 이들 중에서 Hartigan에 의하여 처음 소개된 CHAID는 카이제곱-검정(이산형 목표변수) 또는 F-검정(연속형 목표변수)을 이용하여 다지분리(multiway split)를 수행함으로 데이터를 빠르고 효율적으로 탐색하는 의사결정나무의 기본적인 알고리즘이라고 할 수 있다.

이러한 CHAID 알고리즘을 이용하여 거대한 양의 데이터에 대해 모형을 구축하는 데에는 시간과 노력이 많이 소비되는 단점이 있다. Catlett(1984), Chan과 Stolfo(1993) 등은 의사결정나무 알고리즘에서 샘플링을 사용하면 알고리즘 수행 속도는 감소하나 나무모형의 정확도가 떨어진다고 제시하였다. 이를 확인하기 위하여 본 논문에서는 방대한 데이터 베이스(database;DB)에 대하여 샘플링 기법을 CHAID 알고리즘에 적용시켜 기존의 CHAID 알고리즘의 나무 모형과 동일하면서 모형구축 시간을 단축시키는 알고리즘을 제시하여 그 결과를 탐색하고자 한다. 2절에서는 샘플링에 의한 CHAID 알고리즘을 제시하고, 3절에서는 2절에서 제시한 알고리즘을 바탕으로 예제 및 모의실험을 실시하여 표본추출비율에 따른 수행결과를 탐색하며, 4절에서 결론을 맺고자 한다.

2. 샘플링에 의한 CHAID 알고리즘

전형적인 방법의 CHAID 알고리즘은 아주 방대한 양의 데이터를 대상으로, 의사 결정나무를 형성해나가기 까지 매우 많은 계산과정을 거치게 되므로 많은 시간이 요구된다. CHAID 수행시간을 단축시키는 방법으로 샘플링을 생각할 수 있다. 전체 데이터를 분석하지 않고 전체데이터를 대표할 수 있는 샘플링 된 데이터만 분석한다면 수행시간을 그 만큼 단축시킬 수 있을 것이다. 본 연구에서는 단순임의 추출법을 사용한 CHAID 알고리즘을 제안하고자 한다. 추출한 표본을 이용한 CHAID 알고리즘은 다음과 같다.



<그림 1> 표본추출에 의한 CHAID 수행단계

각 단계에 대한 설명을 요약하면 다음과 같다.

[단계 1] 샘플링

의사결정나무를 형성할 데이터를 만드는 과정이다. 원 DB에서 단순 임의추출법으로 샘플링한 데이터로 새로운 DB를 만든다. 이때 결측치나 불량 데이터는 제거한다. 이에 대한 알고리즘은 다음과 같다.

```
<cfquery name="serch" DataSource="#DB#">
    Select #Prediction_var#
    From #DataBase#
    order by RAND() Limit #random#
</cfquery>
<cfloop query = "serch">
  <cfif #Prediction_var# neq null or 0 or #Prediction_num#>
    <cfinclude template = "reject_data.cfm">
  </cfif>
</cfloop>
```

[단계 2] 분할표

새로 구성된 DB에서 각 예측변수에 대해 목표변수의 범주를 수준으로 하는 분할표 분할표를 만든다. 각 예측변수별로 카이제곱 통계량을 구하고 카이제곱 통계량 DB에 저장한다. 이에 대한 알고리즘은 다음과 같다.

```
<cfinclude template = "delete_partition.cfm">
<cfloop index = i from = 1 to = #ArrayLen(prediction_var)#>
  <cfset partition = ArrayNew(3)>
  <cfloop query=#size#>
    <cfloop index = j from = 1 to = #target_size#>
      <cfinclude template = "create_partition.cfm">
    </cfloop>
  </cfloop>
  <cfset chi_value = Σ(n_ij − υ_ij)^2 / υ_ij>
   <cfset df = 2 * (#size.RecordCount#-1)>
  <cfif #chi_value# gt #Evaluate("chi_statistic[#df#]")#>
    <cfinclude template = "insert_Qvalue.cfm">
  </cfif>
</cfloop>
```

[단계 3] 예측변수 선정

카이제곱 통계량 DB에서 통계량이 가장 큰 예측변수부터 범주의 병합을 시행한다. 예측변수 범주의 각 쌍과 목표변수의 범주를 사용한 부분할표를 만들고, 유의한 정도가 가장 약한 쌍의 유의확률이 주어진 임계값보다 크면, 이들 두 쌍을 하나의 범주로 병합하는 과정을 반복한다. 이에 대한 알고리즘은 다음과 같다.

```
<cfquery name = "tree" DataSource="#DB#">
  select node
  from #partition#
  order by s_value desc
</cfquery>
<cfloop query = "tree">
  <cfinclude template = "sub_partition.cfm">
</cfloop>
```

[단계 4] 부분할표

선택된 예측변수에 대하여 예측변수의 각 쌍과 목표변수의 범주를 사용한 부분할표를 만들고 부분할표에서 구해진 통계량이 카이제곱 통계량보다 크면, 이들 두 쌍을 하나의 범주로 병합하는 과정을 반복한다. 알고리즘은 다음과 같다.

```
<cfloop index = "n" from = 1 to = #sub_size#>
  <cfset partition_sub = ArrayNew(2)>
  <cfset category = ArrayNew(1)>
  <cfinclude template = "create_node.cfm">
  <cfinclude template = "sub_partition.cfm">
  <cfloop query=#size#>
    <cfloop index = j from = 1 to = #target_size#>
      <cfinclude template = "create_sub_partition.cfm">
    </cfloop>
  </cfloop>
  <cfset chi_value = Σ(n_ij − υ_ij)^2 / υ_ij>
  <cfset df = 2 * (#size.RecordCount#-1)>
 <cfinclude template = "Annexation.cfm">
</cfloop>
```

[단계 5] 범주 병합 및 분리

각 예측변수의 부분할표에서 얻어진 범주들의 병합여부를 판정한다. 여기서 부분할표에서 새로 얻어진 예측변수를 사용하여 구해진 통계량이 카이제곱 통계량보다 크면 범주를 병합하고 카이제곱 통계량보다 작으면 예측변수를 범주에 따라 분할한다. 예측변수를 분할한 후 아직 분리되지 않은 예측변수에 대하여 위의 과정들을 반복한다. 이에 대한 알고리즘은 다음과 같다.

```
<cfif #chi_value# gt #Evaluate("chi_statistic[#df#]")#>
  <cfset index_2 = #index_1#-1>
  <cfloop index = "in" from = 1 to = #index_2#>
    <cfif #q_value_sub[1][in]# eq #ArrayMax(q_value_sub[1])#>
      <cfset array_list = "#q_value_sub[2][in]#">
    </cfif>
  </cfloop>
  <cfset category = ArrayNew(1)>
  <cfset new_category = ArrayNew(1)>
    <cfloop index = "id" from = 1 to = #listLen(array_list,'/')#>
      <cfset category[id] = #listGetAt(array_list,id,'/')#>
      <cfset new_category[id] = #listGetAt(array_list,id,'/')#>
    </cfloop>
<cfelse>
   <cfbreak>
</cfif>
```

3. 예제 및 모의실험

본 절에서는 2절에서 구현한 알고리즘을 바탕으로 표본추출비율에 따른 수행시간 및 정확도를 탐색하기 위하여 모의실험을 실시한 후 예제를 적용하였다. 본 실험의 구현환경은 다음과 같다.

CPU : Intel Pentium4-1.8GHz Northwood
RAM : 256MB
O/S : Linux 7.1
Language : coldfusion 5.0
Database : mysql 3.23.51

이 절에서 사용된 데이터는 모 대학교 신입생 1383명을 대상으로 가정환경에 대하여 조사한 실제 데이터로, 이들 중 본 연구에 이용된 변수는 다음과 같다.

* 목표변수 : 가정환경
(1) 화목함 (2) 화목하지 않음
* 예측변수 :
1. 성별
(1) 남자 (2) 여자
2. 아버지학력
(1) 대학이상 (2) 중, 고졸 (3) 초등학교 이하
3. 가정교육방식
(1) 이해심 많음 (2) 보통 (3) 권위적 및 자녀무시 (4) 무관심
4. 생활부담
(1) 부모, 형제, 친척부담 (2) 자신부담(아르바이트) (3) 기타

실제 데이터의 수가 표본추출에 의한 CHAID 알고리즘의 효율성 문제를 논의하기에는 부적절하므로, 먼저 1383건의 데이터에 대하여 random으로 500개의 데이터를 20번 반복으로 추출하여 얻어진 100,000건의 데이터와 40회 반복 추출하여 얻어진 200,000건의 데이터를 이용하여 모의실험을 하였다. 이 모의실험에서 사용된 100,000건의 데이터에 대한 속성은 다음과 같다.

<표1> 100,000건 데이터에 대한 속성

1. 성별

	화목함	화목하지않음
남	34602	17471
여	34706	13221

2. 아버지학력

	화목함	화목하지않음
대학이상	46932	23232
중,고졸	19384	5609
초등학교졸 이하	2992	1851

3. 가정교육방식

	화목함	화목하지않음
이해심 많음	13621	14199
보통	45504	7984
권위적	403	442
무관심	9780	8067

4. 생활부담

	화목함	화목하지않음
부모	60338	24980
자신부담	7397	4614
기타	1573	1098

이들 데이터에 대해 CHAID 알고리즘을 수행한 결과는 <표 2>와 같다.

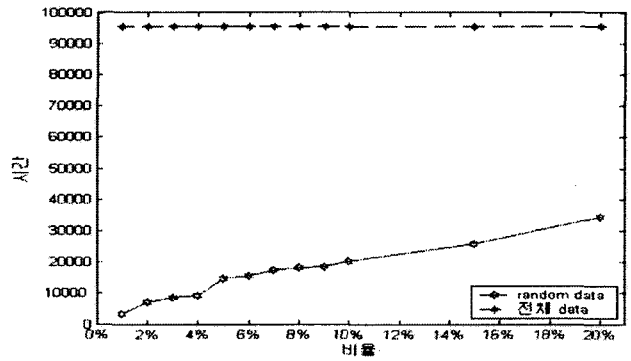
<표 2> 100,000건 데이터에 대한 알고리즘 수행결과(단위 : ms)

구분	전체시간	샘플링제외	샘플size	트리
기존알고리즘	95365	27052	100000	가정교육방식-아버지학력-생활부담-성별
random 20%	34472	15161	20000	기존 알고리즘과 동일
random 15%	25933	14409	15000	기존 알고리즘과 동일
random 10%	20269	12885	10000	기존 알고리즘과 동일(중간노드 약간다름:3번째노드)
random 9%	18617	12042	9000	기존 알고리즘과 동일(중간노드 약간다름:2,3번재노드)
random 8%	18336	11696	8000	기존 알고리즘과 동일(중간노드 약간다름:2,3번째노드)
random 7%	17355	11336	7000	기존 알고리즘과 동일(중간노드 약간다름:2,3번째노드)
random 6%	15713	10903	6000	기존 알고리즘과 동일(중간노드 약간다름:2,3번째노드)
random 5%	14812	10669	5000	기존 알고리즘과 동일(중간노드 약간다름:2,3번째노드)
random 4%	9186	5710	4000	가정교육방식-아버지학력-생활부담
random 3%	9566	6828	3000	가정교육방식-아버지학력-생활부담
random 2%	7139	4858	2000	가정교육방식-아버지학력-성별
random 1%	3342	2088	1000	가정교육방식-아버지학력

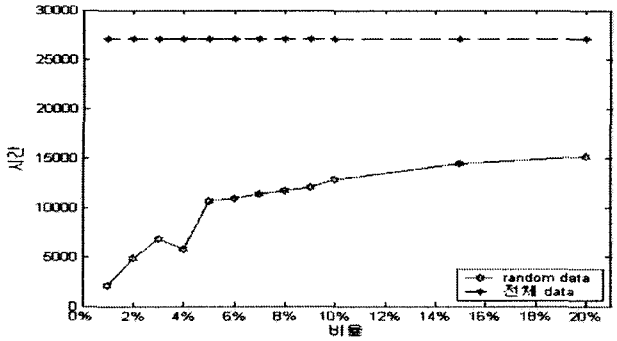
이 표에서 보는 바와 같이 기존의 CHAID 알고리즘 수행시간보다 랜덤 샘플링 알고리즘 수행시간이 현격히 줄어드는 것을 알 수 있다. 기존의 CHAID 알고리즘의 수행시간은 95.365초이고

random 20%일 때와 random 15%일 때는 기존의 CHAID 알고리즘의 트리구조와 동일하면서 34.472초, 25.933초로 알고리즘 수행시간이 1/3, 1/4로 줄어드는 것을 볼 수 있다. 그리고 random 10%에서 random 5%까지는 기존의 CHAID 알고리즘의 트리구조와 별 차이가 없으면서도 수행시간이 1/10 정도로 줄어드는 것을 볼 수 있다. 랜덤 샘플링을 이용한 CHAID 알고리즘이 기존의 CHAID 알고리즘 보다 수행시간을 단축시킬 수 있다고 볼 수 있다.

기존의 알고리즘과 표본추출에 의한 알고리즘의 수행시간을 그림으로 비교하면 <그림 1> 및 <그림 2>와 같다.



<그림 1> 전체수행시간에 대한 랜덤 샘플링비교 그래프(100.000)



<그림 2> 데이터 구축 시간을 제외한 랜덤 샘플링 비교 그래프(100,000)

<표 2>의 100,000건의 데이터에 대한 알고리즘 수행결과에서 기존 알고리즘에 대한 트리 구조, random 10%일 때의 트리 구조, 그리고 random 7%일 때의 트리구조를 부록에 제시하였다.

모의실험에서 사용된 200,000건의 데이터에 대한 속성은 다음과 같다.

<표 3> 200,000 데이터에 대한 속성

1. 성별

	화목함	화목하지않음
남	66732	28568
여	69714	34986

2. 아버지학력

	화목함	화목하지않음
대학이상	36188	11912
중,고졸	94991	47109
초등학교졸 이하	5337	4463

3. 가정교육방식

	화목함	화목하지않음
이해심 많음	19902	16498
보통	30625	28275
권위적	84960	18340
무관심	92	48

4. 생활부담

	화목함	화목하지않음
부모	117421	50779
자신부담	16008	10792
기타	3099	1901

<표 4>에서는 200,000건 데이터에 대한 기존 알고리즘과 랜덤샘플링 알고리즘에 대해 수행결과를 나타내었다.

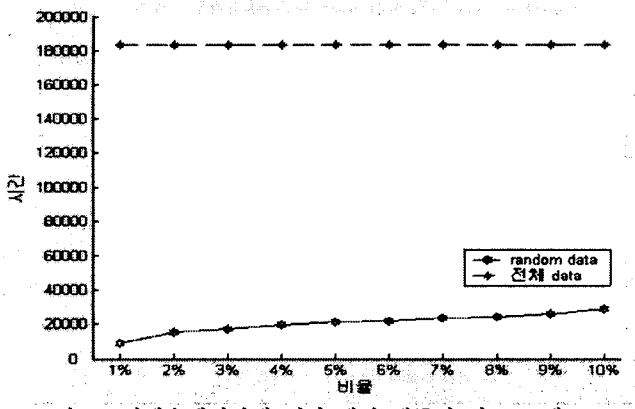
<표 4> 200,000 데이터에 대한 알고리즘 수행표(단위 : ms)

구분	전체시간	샘플링제외	샘플size	트리
기존알고리즘	183398	45922	200000	가정교육방식-아버지학력-생활부담-성별
random 10%	28677	14242	20000	기존 알고리즘과 동일
random 9%	25963	12948	18000	기존 알고리즘과 동일(중간노드 약간다름:3번재노드)
random 8%	24393	12574	16000	기존 알고리즘과 동일(중간노드 약간다름:3번째노드)
random 7%	23966	11648	14000	기존 알고리즘과 동일(중간노드 약간다름:3번째노드)
random 6%	22164	11279	12000	기존 알고리즘과 동일(중간노드 약간다름:3번째노드)
random 5%	21632	10957	10000	기존 알고리즘과 동일(중간노드 약간다름:2번째노드)
random 4%	19861	10247	8000	기존 알고리즘과 동일(중간노드 약간다름:3번째노드)
random 3%	17080	9888	6000	기존 알고리즘과 동일(중간노드 약간다름:3번째노드)
random 2%	15425	9682	4000	기존 알고리즘과 동일(중간노드 약간다름:3번째노드)
random 1%	8808	5278	2000	가정교육방식-아버지학력-생활부담

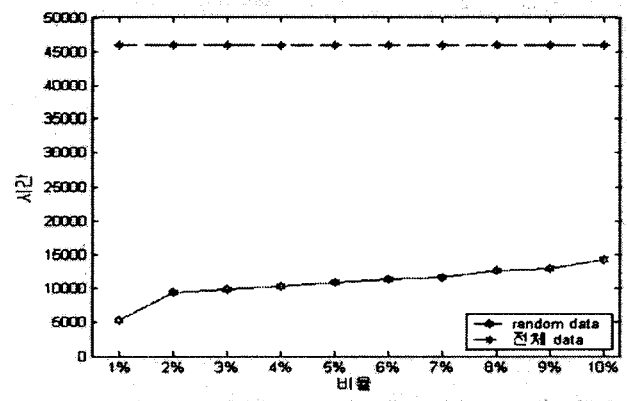
<표 2>에서의 랜덤 샘플링 알고리즘보다 트리의 정확도가 증가한 것을 알 수 있다. <표 2> 랜덤 샘플링 15%일 때가 기존 CHAID 알고리즘과 트리의 형태가 동일하였는데 <표 4>에서는 랜덤

샘플링 10%일때까지 기존 CHAID 알고리즘과 트리의 형태가 동일하다. 이것은 데이터의 크기가 클수록 비례할당 sample size가 적어도 트리의 정확도는 높다는 것을 의미한다.

하지만 랜덤 샘플링은 전체 데이터의 분포를 샘플링에 의하여 잘 대표 할 수 있지만, 잘못된 샘플링을 하였을 때 기존의 CHAID 알고리즘에 의한 트리구조와 차이가 많이 난다.



<그림 3> 전체수행시간에 대한 랜덤 샘플링 비교 그래프(200,000)



<그림 4> 데이터 구축 시간을 제외한 랜덤 샘플링 비교그래프(200,000)

지금부터는 1383명을 대상으로 가정환경에 대하여 조사한 실제 데이터를 이용하여 얻어진 결과를 설명하고자 한다. 이들 데이터에 대한 속성은 다음과 같다.

<표 5> 실제 데이터에 대한 속성

1. 성별

	화목함	화목하지않음
남	476	238
여	484	185

2. 아버지학력

	화목함	화목하지않음
대학이상	270	77
중,고졸	41	25
초등학교졸 이하	649	321

3. 가정교육방식

	화목함	화목하지않음
이해심 많음	133	111
보통	634	111
권위적	188	195
무관심	5	6

4. 생활부담

	화목함	화목하지않음
부모	836	342
자신부담	103	64
기타	21	17

다음의 <표 6>은 실제 데이터를 이용하여 기존의 CHAID 알고리즘과 샘플링 기법들을 사용한 알고리즘과의 수행속도를 비교한 표이다.

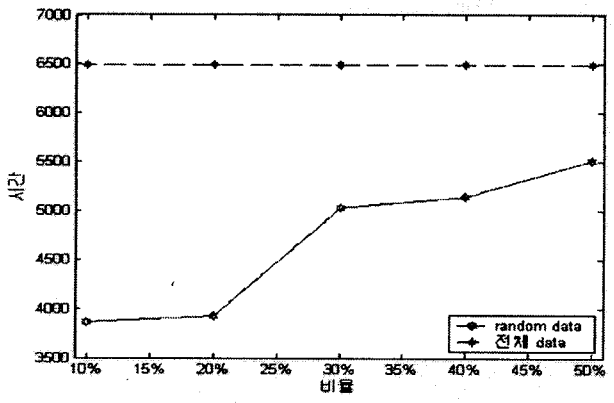
<표 6> 기존알고리즘과 랜덤 샘플링 알고리즘 수행표(단위 : 밀리초)

구분	전체시간	샘플링제외	샘플size	트리
기존알고리즘	6492	2925	1383	가정교육방식-아버지학력
random 50%	5513	2780	692	전체데이터와 동일
random 40%	5141	2673	553	전체데이터와 동일 (중간노드 약간다름:2번재노드)
random 30%	5028	2655	415	전체데이터와 동일 (중간노드 약간다름:2번째노드)
random 20%	3922	1967	277	가정교육방식
random 10%	3865	843	138	가정교육방식

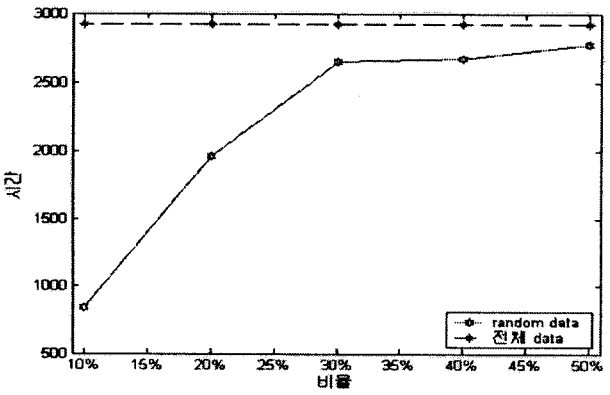
2장에서 제시한 랜덤 샘플링 알고리즘에 의한 CHAID 수행시간이 기존의 CHAID 알고리즘 수행시간보다 현격히 줄어들어야 한다. 하지만 <표6>에서 보는 바와 같이 기존의 CHAID 알고리즘 수행시간과 랜덤 샘플링에 의한 CHAID 알고리즘 수행시간에는 별 차이가 없다.

이 같은 문제점이 발생한 이유는 예제 데이터의 size가 매우 적기 때문이라고 생각한다. 예제 데이터의 크기가 적어서 알고리즘 수행 시간이 실제 알고리즘 계산시간 보다 하드웨어적인 요소에 더욱 민감하게 되었다.

<그림 5>와 <그림 6>은 전체 및 데이터 구축을 제외한 수행시간에 대한 기존 알고리즘의 수행시간과 %별 랜덤 샘플링 수행시간을 비교한 그래프이다.



<그림 5> 전체수행시간에 대한 랜덤 샘플링비교 그래프



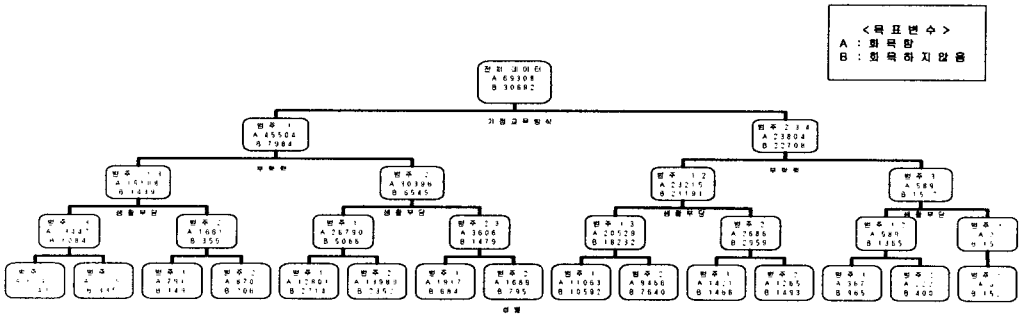
<그림 6> 데이터 구축 시간을 제외한 랜덤 샘플링 비교 그래프

4. 결론

본 논문에서는 의사결정나무 알고리즘 중에서 카이제곱 통계량 또는 F 통계량을 이용하여 다지 분리를 수행함으로써 데이터를 빠르고 효율적으로 탐색하는 의사결정나무의 기본적인 알고리즘인 CHAID 기법에 표본을 이용하는 알고리즘을 제시하고, 이에 대한 정확도와 수행속도를 탐색하였다. 그 결과 표본 추출에 의한 CHAID 알고리즘이 기존의 알고리즘보다 수행속도 면에서 효과적인 것을 알 수 있었으며, 트리 구조면에서도 데이터의 크기가 크고 표본추출비율이 10%이상이면 만족할만한 수준이었다. 반면에 랜덤 샘플링을 사용한 알고리즘은 노드의 정확도 면에서는 문제가 있는 것으로 나타났다. 향후에는 이러한 단점을 보완하기 위한 각종 연구가 진행되기를 기대한다.

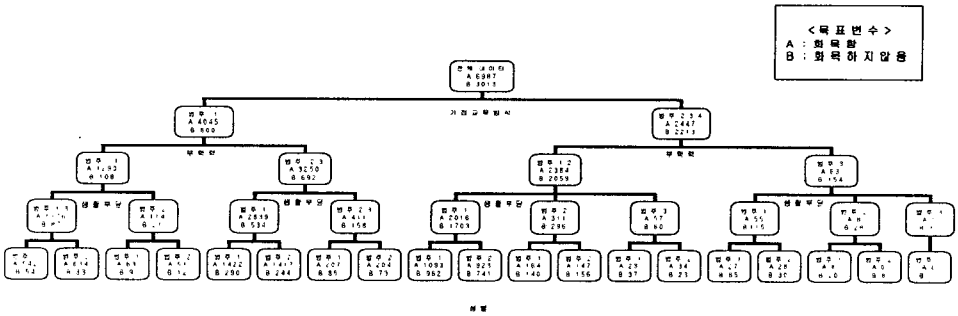
<부록 1>

<전체데이터 트리구조>



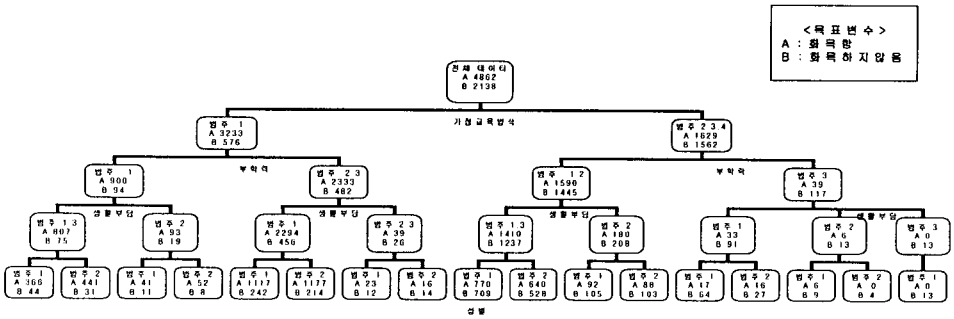
<부록 2>

<Random 10% 트리구조>



<부록 3>

<Random 7% 트리구조>



참고문헌

1. 최종후, 한상태, 강현철, 김은석 (1998). *AnswerTree를 이용한 데이터마이닝 의사결정나무분석*, 서울, 고려정보산업(주).

2. Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984). *Classification and regression trees*, Wadsworth, Belmont.

3. Catlett, J. Megainduction : *Machine Learning on Very Large Databases.* PhD thesis, University of Sydney, 1991

4. Han, j., Kamber, M.(2001) *Data Mining : Concepts and Techniques.* Morgan Daufmann Publishers.

5. Hartigan, J.A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons, Inc.

6. Ji Hyun You(2002). *Clustering algorithm using a Center Of Gravgity for grid-based sample.*

7. Kass, G.(1980), An exploratory technique for investigating large quantities7of categorical data, *Applied Statistics.* 29, 2, 119-127

8. Kass, G.V.(1980), *An Exploratory Technique for Investigating Large Quantities of Categorical Data*

9. Margaret H.Dunham(2003). *Data Mining Introductory and Advanced Topics.*

10. Chan, P.K. and Stolfo, S.J. (1993). Experiments on multistrategy learning by meta-learning. *In Proc. CIKM,*,314-323.

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

12. Quinlan, J.R. (1993), *C4.5 Programs for Machine Learning.* San Mateo, Morgan Kaufmann.

13. Wang, W., Yang, I. and Muntz, R. (1997). *STING: A statistical information grid approach to spatial data mining.* Very Large Data Bases(VLDB'97). 186-195