한국컴퓨터정보학회 동계학술발표대회 논문집 제16권 제2호 (2008. 12)

# 의사결정 트리의 효용성 제고 방안에 관한 비교 연구

석현태
동서대학교 컴퓨터정보공학부
e-mail:sht@gdsu.dongseo.ac.kr

# A Comparative Study on The Effective Use of Decision Tree Algorithms

Hyontai Sug
Division of Computer & Information Engineering, Dongseo University

## 요 약

비교적 적은 크기이면서 예측력에 있어 만족할 만한 의사결정목을 생성하는 방법으로서 적절한 크기의 샘플링을 제안하였다. 일반적으로 샘플의 크기가 작을수록 작은 의사결정목이 생성되므로 적절한 예측 정확도를 갖는 작은 트리를 생성하기를 원할 경우 적당한 크기의 샘플링을 하는 것이 트리의 최적화를 위한 계산을 더 시행하는 것보다 바람직하다고 할 수 있으며, 이와 같은 사실은 현재 알려진 가장 대표적 의사결정목 생성 알고리즘인 C4.5 및 CART를 사용하여 실험으로서 보여주었다.

키워드 : 의사결정목, 샘플링, C4.5, CART

## I. 서론

데이터베이스는 사용함에 따라 자료가 누적되는 것이 일반적이므로 사용하면 할수록 데이터베이스의 크기가 커지게 되는 경향이 있다. 이러한 누적된 많은 양의 데이터에는 발견하면 상당한 가치를 지니게 되는 규칙 혹은 법칙이 있을 수 있는데 이러한 규칙 혹은 법칙을 찾아내는 방법이 데이터마이닝이다. 데이터마이닝의 대상이 되는 데이터베이스는 그 크기가 매우 커서 일반적인 기계 학습적 방법을 바로 적용하기가 곤란한 경우가 많다. 왜냐하면 대부분의 데이터마이닝 알고리듬의 근간이 되는 기계 학습적인 방법은 비교적 소수의 훈련 예를 가지고 만들어진 경우가 많아, 대량의 데이터를 사용한다고 하더라도 찾아낸 규칙의 미래 예측능력이 크게 향상될 가능성이 낮음은 물론 규칙 집합을 찾는데 있어 과도한 계산 시간이 걸려 효용성이 떨어질 수 있기 때문이다. 대표적 데이터마이닝 방법 중의 하나로 그 효과가 잘 알려진 의사결정목의 경우, 일반적으로 기계가 직접 생성한 트리는 의사결정목을 생성하는 데 사용된 데이터의 크기가 커질수록 그 크기가 커지고 이는 결국 해당 의사결정목으로 커버할 수 있는 객체의 숫자를 한정시켜 예측력에 악영향을 미칠 수도 있다. 즉, 오캄의 면도칼법칙[1]에 의하면 간단한 규칙일수록 커버하는 객체의 범위가 넓어져 예측율의 향상에 이바지할 가능성이 높아질 수 있는 것이다. 이밖에도 대형 데이터에 근거해 의사결정목 생성법을 사용해 생성한 트리는 생성시간도 많이 걸리고 그 크기 또한 매우 커서 사람이 바로 참조하기에는 힘든 경우가 많다고 알려져 있다. 따라서 이러한 데이터마이닝 방법을 대형 데이터베이스에 바로 적용하기 보다는 샘플을 이용하거나 적절한 데이터만을 가려 뽑아 사용하는 것이 바람직하다. 본 논문에서는 의사결정목의 효용성을 높이는 측면에서 생성시간을 줄이면서 비교적 작은 크기의 트리를 생성하기 위한 방법으로서 샘플링의 유용성을 실험으로서 보이려한다. 실험은 데이터마이닝의 대표적 방법 중의 하나인 의사결정목에 대해, 그 중에서도 전 세계적으로 가장 잘

알려져 널리 사용되고 있는 방법인 C4.5 및 CART를 가지고 행하였다.

## II. 관련 연구

데이터마이닝의 대상이 되는 데이터는 일반적으로 대형인 경우가 많으므로 대형 데이터에 근거해 의사결정목 생성 알고리즘을 사용해 의사결정목을 생성하는 연구가 많이 행하여졌다. [2, 3, 4] 그러나 이렇게 생성한 트리는 생성시간도 많이 걸리고 그 크기 또한 매우 커서 사람이 바로 참조하기에는 힘든 경우가 많다고 알려져 있다. 따라서 이러한 데이터마이닝 방법을 대형 데이터베이스에 바로 적용하기 보다는 적절한 샘플을 이용하게 되나[5, 6] 샘플은 전체 데이터의 일부에 국한되는 것이므로 여하히 잘 샘플링을 하였느냐가 성공적인 의사결정목의 생성에 많은 영향을 미친다. 현재까지 세계적으로 여러 가지 의사결정목 생성 알고리즘이 제안되었으나 [7] 대표적인 두가지 알고리즘은 C4.5 [8] 및 CART [9]라고 할 수 있다. C4.5는 비교적 짧은 시간에 의사결정목을 생성하나 트리의 크기가 큰 특징을 가지고 있다. 반면 CART는 최적화에 보다 많은 시간을 할애하므로 생성된 트리의 크기가 상대적으로 작다.

## III. 본론

### 1. 의사결정목의 생성 방법

최적의 의사결정목을 만드는 것은 NP-complete 문제이므로 대안으로서 엔트로피 [9]나 지니 [8] 값 등을 기준으로 탐욕 알고리듬(greedy algorithm)에 의해 트리를 생성하게 된다. 대표적 의사결정목 알고리즘의 하나인 C4.5에서는 엔트로피를 의사결정목을 생성하기 위한 각 서브트리의 분지를 위한 기준으로 사용한다. 엔트로피는 다음과 같은 수식에 의해 표현될 수 있다.

$$H(P(X)) = - \sum x \in X \; P(x) \log P(x), \quad \forall x \in X,$$

여기서 P(x)는 x∈X의 확률을 나타내며 사용된 log함수의 밑은 2이다. C4.5 알고리듬에서는 이와 같은 엔트로피에 기초하여 의사결정목을 생성하게 되며, 아울러 엔트로피를 구하기 위한 확률을 계산하기 위해서는 훈련 데이터만으로는 정확한 확률 값을 구할 수 없으므로 의사결정목의 각 잎에 나타나는 훈련 데이터의 수를 헤아린 빈도수에 의해 계산하게 된다.

한편 CART는 Classification And Regression Trees의 약자로 통계적 방법에 의해 의사결정목을 생성하며 C4.5와 함께 대표적 의사결정목 생성 알고리즘의 하나이다. CART에서는 분지를 할 때 다음과 같은 수식의 지니(Gini) 인덱스를 사용하여 가장 순수도가 높은 속성을 기준으로 각 서브트리의 분지를 하게 된다. 여기서 i와 j는 클래스이다.

$$Gini = \sum i \neq j \;\; pi \;\; \log \; pi$$

특정 속성 A에 대한 지니인덱스는 다음처럼 계산된다. 여기서 v는 속성 A의 값을 지정한다.

$$Gini(A) = \sum_v P(v) \sum i \neq j \;\; p(i|v) \; p(j|v)$$

### 2. 가지치기

오캄의 면도칼 법칙에 의해 보다 간단한 형태의 규칙이 미래에도 틀릴 확률이 적으므로 1차로 생성한 트리는 가지치기를 하여 보다 간단한 형태로 만들게 된다. 다음은 C4.5에서 가지치기를 하는 방법이다.

우선, 일단 트리를 끝까지 생성시킨다. 즉, 어떤 단말 노드에서 아직 오분류가 있으면 계속 더 가지를 키워나가고자 서브트리의 루트 속성을 선택하는 분지 작업을 계속하려 할 것인데, 어떠한 남아 있는 속성을 대상으로 엔트로피에 근거한 분지 기준값을 계산해봐도 에러 율의 기대치가 변함이 없으면 서브트리의 생성을 중지시킨다. 그리고 상향식 방식으로 가지를 쳤을 때의 에러율의 기대치를 계산하여 가지를 칠지 말지를 결정한다.

CART에서도 C4.5와 유사하게 진행하나 에러율의 표준오차를 기준으로 판단하게 되며 각 서브트리의 루트노드에 해당 속성 값을 OR 조건으로 포함시키는 특징이 있어 트리의 크기를 줄이는데 상당한 기여를 한다. 그러나 그러한 작업은 상당한 계산시간을 소요하게 되어 C4.5에 비해 계산시간이 많이 걸리는 단점이 있다. 결론적으로 말해 C4.5 알고리즘은 의사결정목을 생성하는 시간은 빠르나 비교적 크기가 큰 의사결정목을 생성하게 되며, CART는 의사결정목을 생성하는 시간은 상대적으로 많이 걸리나 비교적 적은 크기의 의사결정목을 생성하게 된다.

## IV. 성능평가 및 분석

실험에 사용된 데이터는 캘리포니아 대학의 기계학습저장소(maichine learning repository)라는 사이트에 있는 대형 데이터베이스인 census-income 데이터베이스이다.[11] 원래 이 데이터는 1994년 인구조사 자료로부터 추출된 것으로 훈련 예는 총 199,523개의 레코드(99MB)로 구성되어 있으며 이 중 46,716개의 레코드가 중복 또는 서로 모순된 class값을 갖고 있다. 각 레코드는 연소득이 50,000 달러 이상인 50,000+의 클래스와 연소득이 50,000 달러 미만인 50,000-의 두 클래스로 갈라지게 된다. 50,000-가 될 확률은 93.80%이며 50,000+가 될 확률은 6.20%이다. 원 데이터는 일부 속성이 연속치 값을 가지고 있으므로 계산 속도

를 빠르게 하기 위해 미리 이산치로 바꿔주는 전처리 작업을 하였다. 이산치로 바꾸는 방법은 여러 방법들 중 가장 좋은 결과를 내는 것으로 알려진 엔트로피에 기초한 이산화방법 [12]을 사용하였다.

샘플링과 두 의사결정목 생성 알고리즘의 효과를 측정하기 위해 원 데이터의 1/1000, 1/8000 등으로 샘플의 크기를 달리하여 트리를 생성시키고 전체 데이터를 테스트 데이터로 사용하여 에러율을 검정하였다. 실험에 사용된 컴퓨터는 2.33 GHz로 작동하는 인텔 코어2 듀오 CPU를 가진 PC로 메인 메모리 크기는 2GB이다. 다음 표는 각 샘플 크기별 트리의 크기 및 에러율이다. 가지치기를 위한 파라미터 값은 두 방법 모두 디폴트 값을 사용하였다.

표 208. 서로 다른 샘플에 대한 C4.5 의사결정목의 결과

| 구분 | C4.5 (1,000개 샘플) | | |
|---|---|---|---|
| | 트리크기 | 1-에러율 | 생성시간 |
| 샘플 1 | 24 | 94.3 | 0.14 |
| 샘플 2 | 27 | 94.4 | 0.14 |
| 샘플 3 | 22 | 93.9 | 0.08 |
| 평균 | 24.3 | 94.2 | 0.12 |

표 209. 서로 다른 샘플에 대한 CART 의사결정목의 결과

| 구분 | CART (8,000개 샘플) | | |
|---|---|---|---|
| | 트리크기 | 1-에러율 | 생성시간 |
| 샘플 1 | 21 | 94.7 | 131.4 |
| 샘플 2 | 3 | 94.5 | 111.4 |
| 샘플 3 | 9 | 94.6 | 120.7 |
| 평균 | 11 | 94.6 | 121.2 |

C4.5알고리즘은 일반적으로 데이터의 크기에 비례하여 생성된 의사결정목의 크기가 커지는 특성이 있으나 비교적 적은 크기의 샘플을 공급하였으므로 비교적 적은 크기의 의사결정목을 생성한 것을 표1을 통해서 알 수 있다.

CART 알고리즘을 사용한 의사결정목의 실험결과를 정리한 표2의 값을 보면 비록 샘플의 크기가 8배 정도 증가되더라도 에러율의 향상은 그리 크지 않고, 트리를 생성하는데 상대적으로 많은 시간이 소요됨을 알 수 있다. 대부분의 데이터마이닝을 위한 데이터베이스가 아주 큰 것을 생각하면 이상의 실험으로부터 다음과 같은 결론을 내릴 수 있다.

비교적 작은 크기이면서 예측력 혹은 에러율에 있어 만족할만한 의사결정목을 생성하기 위해서는 비교적 적은 크기의 샘플링을 수행하는 편이 바람직하다고 할 것이다. 즉, 샘플의 크기가 작을수록 작은 의사결정목이 생성될 가능성이 높아지므로 이해가 용이한 작고 에러율도 만족할 만한 트리를 생성하기를 원할 경우 상대적으로 비교적 적은 크기의 샘플링을 하는 것이 바람직하다고 하겠다.

## V. 결론

본 논문에서는 비교적 적은 크기이면서 에러율에 있어 만족할 만한 의사결정목을 생성하는 방법으로서 적절한 크기의 샘플링의 유용성을 두가지 대표적 의사결정목 생성 알고리즘을 사용하여 실험으로 확인하였다. 데이터베이스를 대상으로 숨겨져 있는 유용한 지식을 찾아내기 위한 데이터마이닝의 주 대상이 되는 데이터베이스는 그 크기가 매우 큰 경우가 대부분이다. 샘플링을 하지 않고 원 데이터베이스를 그대로 이용할 경우 만일 데이터베이스의 크기가 크면 수행시간이 매우 오래 걸리고 생성되는 트리 또한 일반적으로 매우 크게 된다. 아울러 에러율의 향상도 반드시 기대할 수 있는 것은 아니다. 샘플의 크기가 작을수록 작은 의사결정목이 생성될 가능성이 높아지므로 만족할 만한 에러율의 이해가 용이한 작은 트리를 생성하기를 원할 경우 적당한 크기의 샘플링을 하는 것이 좋다고 할 수 있다. 이와 같은 사실은 원 데이터베이스의 크기가 99MB로 비교적 대형인 census income 데이터베이스와 현재 알려진 가장 대표적 의사결정목 생성 알고리즘으로 의사결정목의 생성속도는 빠르나 그 크기가 상대적으로 큰 의사결정목을 생성하는 C4.5 알고리즘과 의사결정목의 생성속도는 오래 걸리나 그 크기가 상대적으로 작은 의사결정목을 생성하는 CART를 사용해 실험으로서 확인하였다.

## 참고문헌

[1] http://en.wikipedia.org/wiki/Occam's_razor, 2008

[2] J.R.A. Venegas, On Decision Tree Induction for Knowledge Discovery in Very Large Databases, Ph.D. thesis, University of Florida, 1996

[3] J. Catlett., Megainduction: Machine Learning on Very Large Databases, Ph.D. thesis, University of Sydney, Australia, 1991

[4] M. Mehta, R. Agrawal, and J. Rissanen, "SLIQ: A Fast Scalable Classifier for Data Mining", In Proceedings of EBDT, France, March 1996

[5] F. Olken, Random Sampling From Databases, Ph.D. thesis, University of California at Berkeley, 1993

[6] C. Clifton, "Using Sample Size to Limit Exposure to Data Mining", Journal of Computer Security,

vol. 8, pp. 281-307, 2000
[7] P. Tan, M. Steinbach, Y. Kumar, Introduction to Data Mining, Addison Wesley, 2006.
[8] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc., 1993
[9] L. Breiman, et. al., Classification and Regression Trees, Wadsworth International Group, Inc., 1984
[10] J.R. Quinlan, "Induction of Decision Trees", Machine Learning, Vol. 1, pp, 81-106, 1986
[11] P.M. Murphy, and D.W. Aha, UCI Repository of Machine Learning Databases, Technical Report, University of California at Irvine, CA, 1994
[12] H. Liu, F. Hussain, C.I. Tan, M. Dash, "Discretization: An Enabling Techniques", Data Mining and Knowledge Discovery, vol.6, pp. 393-423, 2002.