Decision Tree의 Test Cost 개선에 관한 연구

석현태
동서대학교 인터넷공학부
sht@dongseo.ac.kr

A Study of Improving on Test Costs in Decision Trees

Hyontai Sug
Division of Internet Engineering, Dongseo University

요 약

Decision tree는 목표 데이터에 대한 계층적 관점을 보여준다는 의미에서 데이터를 보다 잘 이해하는데 많은 도움이 되나 탐욕법(greedy algorithm)에 의한 트리 생성법의 한계로 인해 최적의 예측자라고는 할 수가 없다. 이와 같은 약점을 보완하기 위하여 일반적 방법으로 생성한 decision tree에 대하여 다차원 연관규칙 알고리즘을 적용함으로써 짧은 길이의 최적 부분 규칙집합을 구하는 방법을 제시하였고 실험을 통해 그와 같은 사실을 확인하였다.

1. 서 론

Decision tree[1, 2]는 계층적으로 지식을 표현하므로 데이터의 특성을 쉽게 이해할 수 있어 많은 사람들이 선호하는 지식 표현 방법이다. 그러나 비록 decision tree가 많은 사람들에 의해 사용되는 데이터마이닝 방법이라고 하지만 하나의 tree에 모든 데이터에 대한 지식을 표현하려하다 보니 최적의 표현이 되지 못하는 경우가 종종 있다. decision tree를 만들 때 각 부목(subtree)의 루트는 위에서 선택되지 않은 속성 중 선정 기준값이 제일 좋은 속성이 선택되는 탐욕적(greedy choice) 방법에 의해 결정된다. 아울러 tree가 계속 가지를 뻗어감에 따라 하부의 가지는 점점 더 적은 수의 훈련 예를 갖게되므로 tree의 하부로 내려갈수록 각 가지의 신뢰도는 점점 낮아지게 된다. 따라서 decision tree는 속성 값에 대한 불필요한 검사를 하는 경우도 있는 한편 데이터 집단의 어떤 부분집합에 대해서는 최적의 모델이 아닐 경우도 발생한다.

2. 관련 연구

연관규칙[3]은 본 연구의 두 핵심 기술이다. Lazy decision tree 알고리즘은 최근접(nearest neighbor) 알고리즘으로 훈련 예의 테이블에서 입력 예와 가장 유사한 예를 찾아 예측에 사용한다. lazy라는 명칭은 입력 예를 보고 비로소 tree의 한 path를 만들어본다 하여 붙여진 이름이다. 실험 결과 lazy decision tree는 일반 decision tree 보다 예측의 정확도가 높다고 알려져 있다.[4] 그러나 규칙집합이나 tree등과 같은 정형화된 형태의 지식이 아닌 각 입력 예에 대하여 path를 만들어보므로 왜 그러한 판단을 내리게 되었는지에 대한 이해는 곤란하다. 더구나 많은 데이터마이닝 시스템에서 보듯 대상 데이터베이스의 크기가 매우 크다면 계산 시간 또한 문제가 된다.

3. 본 론

3.1 제안하는 방법

다차원 연관규칙 발견 알고리즘[5]을 적용하여 신뢰성이 높은 규칙을 찾아낸다. 이를 위해서는 데이터베이스의 크기에 따라 적당한 최소지지수(minimum support number)를 입력해준다. 응용분야에 따라서는 사용자는 decision tree의 가지에서보다 신뢰성이 높으면서 더 짧은 규칙을 선호할 것이다. 이와 같은 방법에 의해 찾아낸 규칙을 decision tree의 규칙과는 별도로 최적규칙집합이라 부르자. 최적규칙집합에 속하는 규칙은 decision tree보다 더 적은 속성 값 검사로 인해 더 신속한 결정을 내릴 수 있다. 예를 들어, 각 속성이 병의 유무를 판단하기 위해 환자에게 주어지는 검사 항을 나타낸다하고 각각의 검사비용이 다르다고 하자. 의사나 환자는 될 수

있으면 비용이 많이 들어가는 더 많은 검사를 해야하는 규칙보다는 비용이 덜 들어가면서도 신뢰성 있는 결정을 할 수 있는 규칙이 있다면 그와 같은 규칙을 선호할 것이다. 더 간단한 또는 짧은 규칙은 오캄의 면도칼 법칙(Ocam's Razor)에 의해 미래에도 틀릴 가능성이 더 적음으로 인해 선호된다. 다음은 제안하는 방법의 개략적인 절차이다.

1. Decision tree를 생성한다.
2. 다차원 연관규칙 발견법을 적용하여 길이 3 혹은 4의 규칙을 찾는다.
3. 위에서 찾은 연관규칙 중에 decision tree의 각 가지의 부분집합이 되는 규칙을 찾는다.

이러한 규칙은 더 적은 속성값의 검사로 판단이 가능하므로 사용자가 선호할 것이다. 연관규칙을 생성하기 위해서는 특히 iteration의 초기에 후보항목집합의 수가 대폭적으로 줄어드는 장점이 있는 DHP 알고리즘[6]을 응용하여 짧은 규칙을 효율적으로 발견한다. DHP 알고리즘은 원래 트랜잭션 데이터베이스로부터 구매패턴 등을 발견하는데 사용된 1차원 연관규칙 알고리즘인 반면, 우리는 각 항목이 속성 및 값으로 구성된 다차원 연관규칙을 찾게 된다. 또 속성 중의 하나는 판정속성(decision attribute)이고 나머지는 조건속성(condition attribute)인 차이가 있다. 따라서 보다 효율적 수행을 위해 DHP 알고리즘의 후보 항목집합을 생성하는 부분을 수정하였다. 아울러 연관규칙 알고리즘은 명칭 값(nominal values)에만 적용할 수 있으므로 데이터베이스 내의 연속 수치 값(continuous values)은 명칭 값으로 바꿔 주어야 한다. 수치 값을 명칭 값으로 바꿔주는 데는 여러 가지 방법이 다양하게 있으나 decision tree 내에 생성된 구간 나눔과의 호환성을 위해 생성된 decision tree의 분지 기준에 의하여 구간을 정하여 주었다.

3.2 실험

UCI 기계학습보관소의 'census-income'라는 대형 데이터베이스에 대하여 실험을 행하였다. 데이터베이스 내의 총 훈련 예는 199,523개이며 검정 예는 총 99,762개이다. 총 데이터 중 년 소득이 5만 달러 미만인 클래스의 확률은 93.8%이며 5만 달러 이상인 클래스의 확률은 6.2%이다. 데이터베이스 필드 수는 총 41개이며 그 중 8개는 연속 수치 값 속성이다.

Decision tree를 생성하는 데는 C4.5를 사용하였다.[2] 생성된 decision tree의 크기가 너무 크면 이해하기가 힘들므로 전체 훈련 데이터의 1/12 크기의 표본을 사용하여 노드 수 214, 에러율 0.053의 트리를 생성하였다.

8개의 연속 수치 값 속성은 생성된 decision tree의 해당 속성 값에 따른 분지 기준에 의하여 다음과 같은 값을 각 구간의 결점으로 사용하였다.

Attribute	Spliting points
age	29, 36, 48
Capital gains	6849, 7443, 14344
Capital loses	880, 1876
Dividends from stocks	0, 456, 500, 903, 1150
Weeks worked in year	44, 51
Number of persons worked for employer	0, 1
Wage per hour	No interval (not used in the tree)
Instance weight	1792.33, 1829.61, 2079.89, 2207.91, 2306.47

표 2. 연속치 속성 값의 구간치 변환 결점

연관규칙 알고리즘에 의해 생성된 각 규칙에 속하는 훈련 예가 많으면 많을수록 해당 규칙은 통계적으로 더 의미가 있으므로 전체 훈련 예를 가지고 다차원 연관규칙을 생성하였다. 최소지지수는 전체 데이터의 0.2%에 해당하는 399, 신뢰도 95% 이상, 길이 3 이하의 규칙을 생성하였다.

생성된 decision tree의 각 가지에 대한 부분 집합을 이루는 연관규칙을 알아내기 위해 decision tree를 규칙으로 변환시킴으로써 2,681개의 규칙이 만들어졌다. 원 decision tree의 노드 수가 214인데 비해 규칙의 수가 비교적 많이 생성된 이유는 하나의 가지가 여러 규칙으로 전환될 수 있기 때문이다. 예를 들어, {a1, a2, a2+}, 즉, a1≤, a2≤, >a2와 같은 구간 값이 있다하자. 만일 어떤 가지가 {a1, a2}-b1-c1 => d1과 같은 의미를 나타낸다고 하면 이 가지는 a1-b1-c1 => d1 및 a2-b1-c1 =>d1 처럼 결국 2개의 규칙으로 전환되게 된다. 총 2,681 개의 규칙들로부터 신뢰도 95% 이상인

2,556개의 부분집합규칙이 연관규칙 알고리즘에 의해 발견되었다. 예를 들어 다음은 2,681개의 규칙 중의 하나이다.

IF capital gains ≤ 6849 AND
0 < dividend from stocks ≤ 456 AND
capital loses ≤ 880 AND
weeks worked in year ≤ 44 THEN
-50000 (CF=94.7%)

이 규칙에 대하여 다차원 연관규칙 발견법에 의해서 찾아낸 부분 규칙은 다음과 같은 3개이다.

IF capital gains ≤ 6849 THEN
50000 (94.8% with 186003 cases)
IF capital gains ≤ 6849 AND
weeks worked in year ≤ 44 THEN
50000 (95.1% with 93447 cases)
IF capital gains ≤ 6849 AND
capital loses ≤ 880 AND
weeks worked in year ≤ 44 THEN
50000 (95.3% with 183356 cases)

이와 같은 규칙들이 의미하는 바는 비록 decision tree에 에러 율 0.947의 가지가 있다고 하더라도 연관규칙 알고리즘은 그 보다 짧으면서도 신뢰성이 유사하거나 더 높은 규칙을 발견할 수 있다는 것이다. 즉, 보다 적은 속성 값을 검사하더라도 높은 신뢰도로 판단 가능한 속성 및 값은 어떤 것인지를 나타낸다.

4. 결론

Decision tree는 목표 데이터에 대한 계층적 관점을 보여준다는 의미에서 데이터를 보다 잘 이해하는데 많은 도움이 되나 탐욕법(greedy algorithm)에 의한 트리 생성법의 한계로 인해 최적의 예측자라고는 할 수가 없다. 트리가 생성될 때 각 가지는 점점 더 적은 수의 훈련 예를 가지게 되고 따라서 하위의 가지는 상위의 가지보다는 신뢰성이 떨어지게 된다. 따라서 불필요한 속성 값 검사 등을 수행하는 등 최적의 규칙을 만들지 못한다. 앞에서 제안한 방법은 decision tree의 이와 같은 약점을 보완하여 대형 데이터베이스에 대한 데이터마이닝을 보다 효과적으로 하게 하는 여러 가지 이점이 있다. 우선, 연관규칙 알고리즘에 의해 데이터베이스를 철저히 검사함으로써 신뢰성이 높은 짧은 규칙을 발견할 수 있으며, 둘째로, 검사 비용의 절약을 생각할 수 있다. 다시 말해서 만일 최적 규칙 집합에 있는 보다 짧은 규칙에 의해 그 클래스의 예측이 가능하다면 나머지 속성 값은 검사할 필요가 없다는 것이다.

참고문헌

[1] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees, Wadsworth International Group, Inc., 1984

[2] J. Quinlan. C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc., 1993

[3] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I., "Fast Discovery of Association Rules," In *Advances in Knowledge Discovery and Data Mining*, Fayyad, U.M., Piatetsky-Shapiro, G., Smith, P., and Uthurusamy, R. ed., AAAI Press/The MIT Press, pp.307-328, 1996

[4] J. Friedman, R. Kohavi, and Y. Yun. Lazy Decision Trees, AAAI-96, pp.717-714, 1996

[5] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, Inc., 2000

[6] J. Park, M. Chen, and P. Yu. Using a hash-based method with transaction trimming for mining association rules, IEEE Transactions on Knowledge and Data Engineering, 9(5):813-825, Sept.1997