# ID3를 활용한 데이터 마이닝

석 현 태
동서대학교 인터넷공학부

# Data Mining using ID3

**Sug, Hyontai**
Division of Internet Engineering, Dongseo University

## 요약

현재 전세계적으로 데이터마이닝을 위해 많은 종류의 알고리즘이 사용되고 있으나 사용되는 알고리즘의 정확한 이해 없이는 데이터마이닝 결과를 올바르게 해석할 수 없다. 이러한 측면에서 중요한 의사 결정목 생성 알고리즘의 하나인 ID3의 원리를 다루었고, 이를 실세계에서 가장 널리 사용되고 있는 관계형 데이터베이스에 성공적으로 적용하기 위한 훈련 예의 생성 방법 및 연속치를 취급하는 방법을 제시한다.

## Abstract

There are many kinds of algorithms used for the purpose of data mining.. But without the understanding the underlying principles in the algorithm, the result of the data mining cannot be interpreted correctly. In this paper, the principle of ID3 algorithm is explained for that purpose. In addition, the way how to generate good training examples from the relational database is treated, as well as how to convert continuous values into discrete values is considered to use the algorithm for the data mining of real world database.

## I. 서론

데이터베이스를 대상으로 숨겨져 있는 유용한 지식을 찾아내는 방법이 데이터마이닝이다. 이를 위해 의사결정목[1, 2], 신경망[3], 유전자 알고리즘[4] 등 많은 복잡한 방법이 적용되고 있다. 그러나 어떤 방법을 적용하여 데이터마이닝을 하든 그 방법에 대한 정확한 이해가 없는 한 발견된 지식에 대한 올바른 이해를 할 수가 없다. 왜냐하면 이와 같은 방법들은 한정된 시간 내에 답을 찾아내기 위하여 최적해와 가까운 근사해를 찾는 방법에 의존하기 때문이다. 따라서 본 논문에서는 가장 보편적으로 많이 사용되고 있는 데이터마이닝 방법인 의사 결정목(decision tree)을 이해하기 위하여 이를 생성하는 전통적인 방법 중의 하나이면서도 비교적 쉽게 그 원리를 이해할 수 있는 ID3 [5]의 원리를 설명하고 이 알고리즘을 오늘날 데이터베이스로 가장 많이 구축되어 있는 관계형 데이터베이스에 적용하여 데이터 마이닝을 하는 방법을 논하겠다. 앞으로 2장에서는 ID3의 원리를 설명하고, 3장에서는 실세계 데이터베이스에 알고리즘을 적용하는 방법을, 그리고 마지막으로 4장에서는 결론을 제시할 것이다.

## II. ID3의 원리

### 1. 의사 결정목의 생성 방법

ID3는 호주의 시드니 대학에 재직하고 있던 J. Ross Quinlan에 의해 개발되었다. 최적의 의사 결정목을 만드는 것은 NP-complete 문제이므로 대안으로서 엔트로피[1] 나 GINI[2] 값 등을 기준으로 분류했을 때의 각 노드에 속하게 되는 훈련 예의 purity를 살펴, 탐욕 알고리듬(greedy algorithm)에 의해 트리를 생성하게 된다. 여기서 purity란 해당 노드에 얼마나 같은 클래스의 훈련 예가 소속되어 있나를 뜻한다. 따라서 같은 클래스의 훈련 예가 해당 노드에 많이 모여 있으면 있을수록 purity가 높게 된다. 이 중 ID3에 사용된 엔트로피 measure의 purity 검사 방법에

대해 살펴보자. 샤논(Shannon)은 엔트로피를 다음처럼 정의한다.

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

단, P(x)는 x∈X의 확률을 나타낸다. 여기서 log함수의 밑은 2이다.

예를 들어, 각 원소의 일어날 확률이 1/3인 확률변수 X={x1, x2, x3}가 있다고 하자. 확률변수 X에 대한 불확실성, 즉 엔트로피는

$$H(P(X)) = -\{(1/3)\log(1/3) + (1/3)\log(1/3) + (1/3)\log(1/3)\} = 1.6$$

만일 각 원소의 일어날 확률이 위와는 다르게 x1 = 1/2, x2 = 1/2, x3 = 0이라면 엔트로피는 다음처럼 계산된다.

$$H(P(X)) = -\{(1/2)\log(1/2) + (1/2)\log(1/2) + 0\} = 1.$$

이 두 값들을 비교해보면 전자의 확률분포가 후자보다는 더 불확실성이 높다는 것을 알 수 있다. 다시 말해 후자는 셋 중에 하나는 확실히 일어나지 않을 것을 알고 나머지 둘 중 하나가 일어날 것임을 아는 데 비해, 전자는 셋 중 하나가 일어날 것임을 앎으로 후자가 좀더 많은 정보를 가지고 있다고 볼 수 있다.

또 다른 예로, 하나는 균형이 잘맞는 동전이고 다른 하나의 동전은 좀 구부러져 있어 앞면이나 뒷면이 나올 확률이 다르다고 하자. 즉 균형이 잘 맞는 동전은 앞면과 뒷면이 나올 확률이 같고, 균형이 맞지 않는 동전은 앞면과 뒷면이 나올 확률이 각각 9/10, 1/10이라고 하자. 균형이 잘 맞는 동전의 엔트로피는

$$-\{(1/2)\log(1/2) + (1/2)\log(1/2)\} = 1,$$

한편 균형이 맞지 않은 동전의 엔트로피는

$$-\{(9/10)\log(9/10) + (1/10)\log(1/10)\} = 0.4691$$

로 계산할 수 있다. 따라서 균형이 맞는 동전은 앞면이나 뒷면이 나올 확률이 같아 다음에 어떤 면이 나올지 예측하기 힘든 반면, 균형이 맞지 않은 동전은 앞면이 나올 확률이 높아 예측이 비교적 용이하다고 할 수 있다. 즉 엔트로피 값이 작을수록 더 확실한 정보라고 할 수 있다. 이와 같은 엔트로피에 기초하여 Quinlan의 ID3 알고리듬[7]에서는 의사결정목을 다음처럼 생성시킨다.

각각의 확률이 $q_1, q_2, \ldots, q_n$인 r개의 class가 있다고 하고, 속성 A가 n개의 서로 다른 값을 가질 수 있고 각 값이 나타날 확률이 각각 $p_1, p_2, \ldots, p_n$이라고 하자. 그리고 각 값의 각 class에 대한 확률을 각각 $p_{11}, p_{12}, \ldots, p_{1r}, p_{21}, p_{22}, \ldots, p_{2r}, \ldots, p_{n1}, p_{n2}, \ldots, p_{nr}$이라고 하자. 속성 A의 한 값 $a_i$에 대한 기대정보량 혹은 엔트로피는

$$- \sum_{j=1\sim r} \; p_{ij} \log p_{ij}$$

따라서 속성 A에 대한 기대정보량은 다음과 같다.

$$\mathrm{info}_A(T) = \sum_{i=1\sim n} \; p_i \; (- \sum_{j=1\sim r} \; p_{ij} \log p_{ij} \; )$$

여기서 T는 전체 모수에 대한 확률변수이다. 그런데 위 식은 값이 작을수록 정보량이 많다는 뜻이 되므로 양수로 만들어 생각을 편하게 하기 위해

$$\mathrm{info}(T) = - \sum_{i=1\sim r} \; q_i \log q_i$$

를 피감수로 하여 속성 A에 대한 정보량을 빼서 사용하며, 이것을 속성 X의 gain이라 부른다. 즉,

$$\mathrm{gain}(X) = \mathrm{info}(T) - \mathrm{info}_A(T)$$

$$= (- \sum_{i=1\sim r} \; q_i \log q_i \; ) \; - \; ( \sum_{i=1\sim n} \; p_i \; (- \sum_{j=1\sim r} \; p_{ij} \log p_{ij} \; ))$$

확률 값에 대해서는 실제 모수를 구하기는 현실적으로 불가능하므로 대신 각 노드에서 각 값이 나타난 횟수에 근거한 frequency ratio를 사용한다. 이것은 우리가 근사해 밖에 구할 수 없는 이유이기도 하다.

## 2. 트리 생성의 예

다음 페이지의 표1과 같은 훈련 예를 가진 데이터 테이블이 있다고 하자. 테이블에서 A, B는 조건 속성이다. 9개의 양의 class와 5개의 음의 class로 데이터가 구성되므로

$$\mathrm{info}(T) = -(9/14)\log(9/14) \; -(5/14)\log(5/14) = 0.94$$

속성 A의 경우 5개의 0은 2개의 양의 class 및 3개의 음의 class로 구성되고, 4개의 1은 4개의 양의 class로 구성되는 한편, 5개의 2는 3개의 양의 class 및 2개의 음의 class로 구성되므로

$$\mathrm{info}_A(T) = 5/14\{-(2/5)\log(2/5) \; -(3/5)\log(3/5)\} \; +$$

$$4/14\{-(4/4)\log(4/4) -(0\log 0)\} + 5/14\{-(3/5)\log(3/5)\} = 0.694$$

$$\text{split\_info}(A) = -(5/14)\log(5/14) -(4/14)\log(4/14) -(5/14)\log(5/14) = 1.5774$$

| A | B | class |
|---|---|---|
| 0 | 0 | + |
| 0 | 0 | - |
| 0 | 1 | - |
| 0 | 1 | - |
| 0 | 1 | + |
| 1 | 0 | + |
| 1 | 1 | + |
| 1 | 0 | + |
| 1 | 1 | + |
| 2 | 0 | - |
| 2 | 0 | - |
| 2 | 1 | + |
| 2 | 1 | + |
| 2 | 1 | + |

표 1. 데이터 테이블

따라서,

$$\text{gain}(A) = 0.94 - 0.694 = 0.246$$

유사하게 속성 B에 대해 구해보면

$$\text{info}_B(T) = 6/14\{-(3/6)\log(3/6) -(3/6)\log(3/6)\} + 8/14\{-(6/8)\log(6/8) -(2/8)\log(2/8)\} = 0.892$$

$$\text{gain}(B) = 0.94 - 0.892 = 0.048$$

따라서 A의 gain 값이 B의 gain 값보다 크므로 A를 의사결정목의 루트로 정하게 된다. 이렇게 생성된 의사결정목은 다음과 같다:

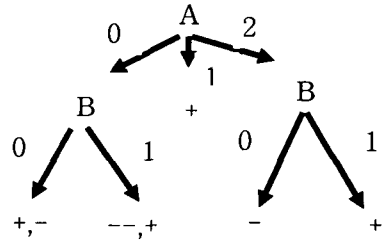


그림 1. A를 루트로 선택할 때의 의사 결정목

그림 1에서 단말 노드를 보면 클래스 값으로 +나 -가 표시되어 있는데 부호가 하나만 있는 것은 그 쪽으로 분류된 훈련 예는 100% 모두 해당 클래스로 속하는 것을 표시한다. 맨 왼쪽의 단말 노드의 + 및 -는 표1의 값에서 알 수 있듯이 하나는 +로 다른 하나는 -로 분류되었음을 뜻한다. 왼쪽에서 두 번째 노드는 -로 두 개의 훈련 예가, +로 하나의 훈련 예가 분류되었음을 알 수 있다. 따라서 미래에 예측해야될 데이터에 대해 만일 맨 왼쪽의 단말 노드로 가게된다면 그 것은 + 또는 -의 클래스로 예측할 수 있는 데 그 확률은 각각 50%라는 의미이다.

아래의 표2는 속성 B가 트리의 루트로 선택되었을 때의 의사 결정목을 살펴보기 위해 속성 B를 기준으로 표 1을 다시 정리한 것이다. 만일 속성 A를 루트 노드로 선택하지 않고 속성 B를 선택하였다면 다음의 그림 2와 같은 의사결정목이 생성될 것이다.

| B | A | class |
|---|---|---|
| 0 | 0 | + |
| 0 | 0 | - |
| 0 | 1 | + |
| 0 | 1 | + |
| 0 | 2 | - |
| 0 | 2 | - |
| 1 | 0 | - |
| 1 | 0 | - |
| 1 | 0 | + |
| 1 | 1 | + |
| 1 | 1 | + |
| 1 | 2 | + |
| 1 | 2 | + |
| 1 | 2 | + |

표 2. 데이터 테이블

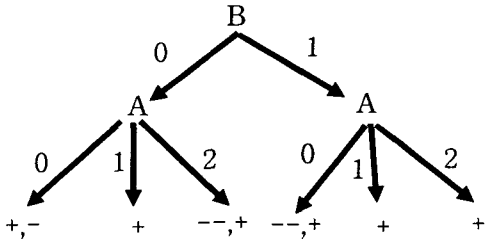


그림 2. B를 루트로 선택했을 때의 의사 결정목

그림 1과 2를 비교하면 단말 노드의 purity 측면에서는 유사하나 그림 1의 트리가 단말 노드 수가 1이 작음을 알 수 있고 특히 속성 A의 값이 1일 때는 하위 노드로 내려가지 않고 바로 클래스를 판정할 수 있으므로 좀 더 우수한 의사 결정목이라 할 수 있다.

마지막으로 표 1이나 2의 훈련 예와 달리 만일 속성들의 수가 더 많다면 각 부목에 대해서도 계속적으로 유사한 작업을 하여 트리를 생성하게 된다.

## III. 실세계 데이터베이스에 대한 적용

### 1. 관계형 데이터베이스에 대한 적용

관계형 데이터베이스는 정규화된 여러 테이블로 구성되어 있다. 데이터 마이닝을 하기 위해서는 이러한 정규화된 테이블들을 조인하여 표 1의 데이터와 유사한 형태의 테이블을 만들게 된다. 일반적으로 정규화된 테이블은 제3정규형의 형태이나 조인을 하면 제3정규형의 성질, 즉 제2정규형이면서 키가 아닌 속성들간에 상호 의존성이 없다는 특징을 깨뜨릴 수도 있다. 그러나 의사결정목의 대상이 되는 자료의 각 속성은 오직 클래스 속성에만 의존성이 있고 나머지 속성들 간에는 함수적 종속관계를 가져서는 안 된다는 제약 조건이 있다. 따라서 조인 작업을 할 때에는 이 점에 유의하여 조인하여야 한다. 이를 위해 경우에 따라 프로젝션 및 조인의 적절한 배합이 필요하다. 즉 조인을 통해 생성된 결과 테이블에서 함수적 종속관계가 있는 속성들이 섞여 있으면 프로젝션을 통해 함수적 종속관계에 있는 그들 속성들 중에 비교적 덜 중요하다고 생각되는 속성을 제거해 주어야 한다.

### 2. 연속치에 대한 처리

ID3는 기본적으로 이산치 자료를 대상으로 트리를 생성할 수 있다. 따라서 연속치 속성의 자료는 미리 이산치로 변환시켜주어 입력하여야 한다. 일반적으로 연속치를 이산치로 바꾸는 다양한 방법이 존재한다. 논문 [6]에서는 8가지 각기 다른 이산화 방법을 비교하여 가장 바람직한 방법으로 엔트로피에 기초한 방법을 추천하고 있다. ID3가 트리를 생성하기 위해 엔트로피에 기초하고 있는 만큼 방법의 일관성 측면에서 엔트로피에 기초한 변환 방법이 바람직하다하겠다. 자세한 변환 방법은 논문 [6]을 참조하기 바란다.

## IV. 결론

본 논문에서는 데이터 마이닝에 대한 정확한 이해의 일환으로 ID3 알고리즘의 원리 및 이를 실세계 데이터베이스에 적용할 때의 고려할 점을 논하였다. ID3 알고리즘은 의사 결정목 생성에 있어 여타 다른 알고리즘에 많은 영향을 준 알고리즘이다. 따라서 이에 대한 올바른 이해는 다른 알고리즘을 보다 쉽게 이해하고 또한 발전시킬 수 있는 기초가 된다 하겠다. 아울러 ID3 알고리즘이 기계학습을 위해 발명된 알고리즘인 만큼 이를 오늘날 가장 많이 사용하고 있는 관계형 데이터베이스에 어떻게 적용하여야 올바른 데이터마이닝을 할 수 있을 지 살펴보았다. 즉 단말 노드를 제외한 의사 결정목의 각 노드에서 사용되는 속성들간에는 상호독립성이 보장되어야 하며 오직 클래스 속성만에 모든 속성이 의존성을 갖는 지를 확인해야 하는 절차를 조인과정에서 거치도록 하는 것이다. 마지막으로 연속치 속성을 다루는 방법을 살펴보았다.

## 참고문헌

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

[2] Breiman, L. et. al., *Classification and Regression Trees*, Wadsworth International Group, Inc., 1984

[3] Fu, L.M., Neural Networks in Computer Intelligence, McGraw Hill, Inc., New York, 1994

[4] Dejong, K,A., Spears, W.M., Gordon, D.F., Using Genetic Algorithms for Concept Learning, Vol.13, 1993, 161-188

[5] Quinlan, J.R., *Induction of Decision Trees*, Machine Learning, Vol. 1, pp, 81-106, 1986

[6] Liu, H., Hussain, F., Tan, C.I., Dash, M., Discretization: An Enabling Techniques, Data Mining and Knowledge Discovery, Vol.6, 393-423, 2002