

# 프로토타입 학습 모델에 관한 연구 (A Study on a Prototype Learning Model)

송 두 현\*  
(Doo-Heon Song)

## 요 약

우리는 개념 학습에 있어서 전통적으로 사용되어 온 연역 트리 구성법이나 규칙 학습법과 다른 새로운 개념 표현 기법을 소개하고자 한다.

우리의 PROLEARN 알고리즘은 각 클래스로부터 주어진 예제를 가장 잘 설명할 수 있는 가상 예제, 즉, 프로토타입을 하나 이상 학습하고 이것을 마치 주어진 예제처럼 취급하여 일반적인 개체 중심 학습법처럼 분류하도록 한다. 우리의 프로토타입 개념은 인지 심리학에서 사용한 같은 용어와는 하나의 개념이 하나 이상의 프로토타입을 가질 수 있도록 한 점에서 다르며 학습된 프로토타입은 근본적으로 '가상 예제'라는 점에서 다른 개체 중심 학습법과 다르다. 실험 결과 이 알고리즘은 정확도에서 다른 알고리즘에 뒤지지 않으며 실제 학습 문제에서 자주 발생하는 불안정성 문제, 즉 훈련 예제 집합이 바뀌면 알고리즘의 정확도도 영향받는 부분도 해소하였다.

## ABSTRACT

We describe a new representation for learning concepts that differs from the traditional decision tree and rule induction algorithms.

Our algorithm PROLEARN learns one or more prototype per class and follows instance based classification with them. Prototype here differs from psychological term in that we can have more than one prototype per concept and also differs from other instance based algorithms since the prototype is a "fictitious ideal example". We show that PROLEARN is as good as the traditional machine learning algorithms but much more stable than them in an environment that has noise or changing training set, what we call 'stability'.

## 1. 서론

기계 학습의 분류(classification) 문제를 해결하고자 하는 다양한 알고리즘을 구분하는 한 가지 방법은 그 알고리즘이 학습 목표 개념  $C$ 를 결과적으로 기술하는데 있어 可用한 모든 속성(feature)의 특징한 부분집합을 이용하는가 아니면 모든 속성을 이용하는가 하는 것이다.

이것은 또 개념 학습이 그 결과를 속성의 조합으로 설명할 수 있어야 하는가 아니면 단순히 알고리즘이 학습의 결과로써 주어진 예제를 정확하게 분류하기만 하면 되는가 하는 문제와 맞물려져 여러 가지 다른 유형의, 그러나 비슷한 정도의 정확성을 갖는 다양한 알고리즘들을 지난 10여 년 간 생성해 왔다.

\* 정희원 : 용인송담대학 컴퓨터소프트웨어과 교수

C4.5[3], CN2[10]으로 대표되는 MDL(Minimum Description Length [12])를 기반으로 한 알고리즘은 예제를 Divide-and-conquer를 통해 분리(partition)해 내고 그 과정에서 얻어지는 트리 구조의 필수 불가결한 속성과 값의 부분집합을 통해 개념을 설명하거나 [10], separate-and-conquer를 이용, 예제를 연속적으로 재분리하면서 분류법칙(classification rule)을 생성한다[3]. 비교적 빠른 속도로 문제를 해결하는 이群的 알고리즘은 그러나 Exclusive-OR같은 문제의 해결에 난점이 있거나 약간의 부정확한 자료(noisy data)에 의해 개념 학습에 지장을 받는 문제점이 지적되어 왔으며 실제 문제에 응용되었을 때 그 분야의 전문가들에 의해 부정확한 이론의 추출이라는 비판을 받아 왔다. 이러한 단점들은 이 알고리즘들의 기반이 되는 MDL이 분류에 가장 효과적이라고 사료되는 속성만을 고려하므로 속성간의 연합에 의한 개념 형성을 무시하는 데에 그 이유가 있다.

Bayesian Classifier[5], 뉴럴 네트워크[11], 예제 중심 분류법(Instance based Algorithms : [1], [4])등은 모든 속성을 고려하고 주어진 예제들로부터 분류에 필요한 확률 도표 혹은 잠재적 개념의 범위를 형성한다. 실험을 통하여 이 群的 알고리즘이 앞의 MDL based 알고리즘 못지 않은 정확성을 유지하며 그 기준이 어떠한 엉뚱한 이론 추출이 없다는 점에서 안정성(stability)있는 알고리즘으로 평가받는 반면 학습된 개념을 사람이 이해할 수 있는 형태로 재기술(redescribe)할 수 없다는 점에서 전문가 시스템 작성이나 문제해결 시스템(Problem solving system)들에서는 환영받지 못하였다.

컴퓨터 기계학습에서는 심각하게 다루어지지 않았지만 심리학에서는 인간과 동물이 프로토타입(prototype)을 학습한다는 여러 가지 실험적 증거들이 있다[2][6][7]. 여기서 prototype이란 학습 목표개념을 설명할 수 있는 理想的예제(ideal example)를 말하며 심리학에서는 각 개념 당 한 개의 prototype만이 존재한다고 믿었다. Prototype은 이 경우 주어진 예제 집합 중 어느 하나일 수도 있고 그 중에는 포함되지 않은 가상적 예제일 수도 있다. Prototype이 가상적 예제라는 것이 여기에서 의미를 갖는데 아무리 주어진 예제 집합이 크더라도 그것이 현실 세계의 sample이며 그 sample을 통하여 학습 목표를 추정해 내는 것이므로 기계 학습의 가정에도 합치된다. 우리

가 컴퓨터로 이상적 예제인 prototype을 학습한다면 그 형태는 전술한 기계 학습의 두 진영의 중간 형태가 될 것인데 이는 prototype이 모든 속성을 고려하지는 않지만 중복되는 속성을 무시하지도 않으며 다만 학습 목표와 무관(irrelevant)하다고 여겨지는 것을 제외하되 결과물인 개념은 속성과 값의 조합으로 주어지므로 정확성과 안정성, 속도 면에서 다른 두 진영의 장점을 조합하게 될 것이다.

즉, 이 논문에서 우리가 소개할 prototype은 분류에 필요한 최소한의 속성의 부분집합(MDL 진영)만이 아니라 그 개념을 가장 잘 설명하는 이상적 예제가 포함하는 有用한 중복적 속성(informative redundancy)을 허용한다. 또 이것이 심리학적 정의에 입각한 가상적 예제라는 점에서 예제 중심 분류법(Instance based algorithm)에 typicality 기준을 적용한 Zhang(1992)의 연구와 다르며[14](Zhang은 주어진 예제 중 가장 typical한 예제를 이 기준을 적용하여 선택했다.) 이에 덧붙여 우리는 심리학자들과 다른 견해를 또한 표방하고자 한다. 즉, 심리학자들은 한 개념에 오직 한 개의 prototype이 존재한다고 믿었기 때문에 매우 제한된 문제에서 그것이 존재하고 학습될 수 있음을 보였지만 기계학습의 유명한 문제 중의 하나인 Exclusive-OR만 하더라도 2개의 prototype이 동시에 존재하여야 하므로 심리학자들의 이 가정은 너무나 경직된 것이라 할 것이다. 따라서 우리는 학습 목표인 개념은 하나 이상의 prototype을 가질 수 있도록 학습 알고리즘을 설계하였다.

이 논문의 제 2장에서는 우리가 개발한 PROLEARN(PROTOTYPE LEARNING)을 설명하고 제 3장에서는 이 알고리즘의 실험적 논거를 UCI실험 자료집[8]의 여러 문제를 통해 증거하며 제 4장에서 결론을 맺도록 하겠다.

## 2. PROLEARN 알고리즘

### 2.1 용어 정의

이 논문 전체를 통하여 주어진 예제집합  $E = \{e_i\}$ 는 속성의 집합  $F = \{f_j\} 1 \leq j \leq n$ 와 그 값  $V_j = \{v_{jR}\}$ 로 표시되며 각  $e_i$ 는 N class중의 하나로 분류된다.

$C = \{c_i \mid 1 \leq i \leq N\}$ 은 class 집합을 나타낸다. Prototype 은 class 정보와 속성과 그 값의 결합의 불리안 곱으로 나타낸다. 즉,

$$P = \{ (c_i, ch) \mid c_i \in C, ch = \bigwedge_{f_j \in F} (f_j \text{ op } v_{jR}, S_j) v_{jR} \in V_j, S_j$$

s는 strength, op는 관계연산자,

$F' \subseteq F, F'$ 는 학습목표와 유관한 속성의 집합이며 학습의 결과인 Prototype을 각 class마다 하나 이상의 P를 가질 수 있으므로 Prototype = {P<sub>i</sub>}로 표시된다. 일례로, 예제 E<sub>i</sub>가 4개의 부울대수 변수  $f_1, f_2, f_3, f_4$ 로 표시되고 학습 목표가  $f_1$ 과  $f_2$ 의 exclusive-OR일 때 학습의 결과는 Prototype = ((Class = +, Features : ( $f_1 = T, 0.85$ ), ( $f_2 = T, 0.9$ )), (Class = +, Features : ( $f_1 = T, 0.9$ ), ( $f_2 = T, 0.8$ )))로 나타낸다.

## 2.2 PROLEARN(PROtotype LEARning) 알고리즘.

PROLEARN은 주어진 예제 집합으로부터 개개의 class에 관한 Prototype을 학습한다. 전술한 대로 한 Class에서 두 개 이상의 Prototype 표현이 생성될 수 있으므로 PROLEARN은 전체 예제 집합 중 현재 학

습중인 class  $C_i$ 로 표시된 부분 집합 E로부터 top-down으로 partition을 생성한다. partition을 생성하는 방법은 주어진 속성의 집합  $F_j$ 와 그 값  $V_{jR}$ 의 연합, 즉,  $E_{jRi} = \{e_i \mid F_j = v_{jR} \text{ in } e_i\}$ 를 구한 후 이 부분 집합들 중 이항 분포 혹은 정규분포를 통해 볼 때 생성 우연 지수(randomness of the feature-value subset)가 가장 낮은, 즉, 그 집합이 우연히 생성된 것이 아니라 결정적 공통점을 가졌을 확률이 가장 높은 것을 선택한다.

이것을 결정하는 기준을 Quality( $E_{jRi}$ )라고 두자. 이 부분 집합은 그러나 전체 집합  $E_i$ 의 quality보다도 통계적으로 유의하게 높아야 한다(지표 Quality가 높으면 subset의 randomness는 낮아지도록 설계되었다). 이렇게 하나의 유용한 Partition이 생성되면 선택된  $F_j$ 를 제외하고 다시 이 과정을 반복하여 더 이상 그러한 유용한 partition이 없을 때까지 계속한다. 이렇게 생성된 각 partition은 이제 무관한 속성 (irrelevant features)을 제거하고  $F_j$ 마다 가장 공통적인 값을 취하도록 하면 그것이 이 partition을 대표하는 이상적 예제, 곧, prototype이 된다. 이 과정이 간략히 [그림 1]에 표시되었다.

$\forall c_i \in C$ , 다음을 수행하라.

```

Let Prototype = { }, E = { ei | class(ei) = Ci }
Repeat
  ∀ Fj & VjR, Fj unused,
  Enumerate EjRi = { ei | Fj = vjR in ei }
  Compute Quality( EjRi )
  Let Q = EjRi such that Quality( EjRi )이 위의 partition중 최대값.
  If Quality(Q) > Quality( Ei )
    then do ;
    set P = Q로부터 무관 Fj를 제거하고 나머지 Fj에서 공통인 값을 추출한 것.
    Prototype = Prototype (합집합표시) U P
  end
Until 더 이상의 유용한 partition EjRi가 존재하지 않을 때
    
```

[그림 1] PROLEARN 알고리즘  
[Fig.1] PROLEARN Algorithm

그러면 여기서 가상적 partition  $E_{jR}$ 의 quality를 계산하는 방법을 살펴본다. 위의 알고리즘에서 보듯이 PROLEARN 알고리즘은 하나의 유용한 partition을 생성할 때마다 하나의 속성  $F_j$ 를 사용하고 그 다음 loop에서는 더 이상  $F_j$ 를 고려하지 않는다.

$$\text{Quality}(E_{jR}) = \sum_j C\text{Max}(F_j)$$

$$F_j' = \{F_j | F_{j\text{unused}}, C\text{Max}(F_j) > n\}$$

$$C\text{Max}(F_j) = \text{Max}(\text{bnp}(N_{jR}, N_i, 1/K))$$

$\text{bnp}(m, n, p)$ 는  $n$ 개의 이항 분포를 따르는 trial 중  $m$ 개 이상이 성공할 확률을 나타내며  $P$ 는 단위 trial이 성공할 확률이다.  $N_{jR}$ 는 주어진 partition  $E_{jR}$  중 속성  $F_j = V_{jR}$ 인 것의 숫자,  $N_i$ 는 class가  $C_i$ 인 것의 숫자  $K$ 는  $F_j$ 가 가질 수 있는 값의 개수를 나타낸다. 따라서  $C\text{Max}(F_j)$ 는 그 partition이 속성  $F_j$ 에 의하여 나누어졌을 때 가장 구성원이 의미 있게 혹은 가장 덜 우연하게 모였을 경우를 나타내며 Quality 지표는 이것을 현재 사용 가능한 모든 속성에 대하여 살펴본 것이다  $F_j'$ 의 정의 안에 있는  $C\text{Max}(F_j) > n$ 의  $n$ 은 실험 parameter로서 이 논문에서는 90%의 신뢰도( $n=0.9$ )를 가정하였다.

### 2.3 분류(classification)

PROLEARN은 기본적으로 예제 중심 분류법과 그 궤를 같이 한다. IBL[1]이나 PEBLS[4]등이 예제 집합 내의 특정한 예제를 선택한 것과 달리 PROLEARN은 학습과정을 통해 가상적인 prototype을 각 class마다 추출해 낸 것이 다르다. 만약 PROLEARN에게 사용자가 어떤 예제를 주고 그 class를 판별하라고 하면 PROLEARN은 다른 예제 중심 분류법 알고리즘과 마찬가지로 그 예제와 prototype간의 의미적 거리를 계산하여 가장 근접한 prototype을 추출한다. 이 prototype의 class가 주어진 예제의 class로 추정된다.

알고리즘마다 서로 다른 거리 계산법이 사용될 수 있으나 우리는 여기서 MVDM[4]의 변형을 차용하였다. 본래의 MVDM은 두 속성 값  $V_1$ 과  $V_2$ 간의 거리계산을 위하여 주어진 전체 훈련 집합(training data)을 사용하였다. 우리는 유용한 prototype이 둘 이상 존

재할 수 있으므로 이를 아래와 같이 개량하였다.

$$d(V_1, V_2) = \sum_{i=1}^R \left| \frac{c_{1i}}{c_1} - \frac{c_{2i}}{c_2} \right|$$

$P$ 는 prototype의 개수

$c_1, c_2$ 는 전체 집합 중 값  $V_1, V_2$ 가 발생한 빈도

$c_{1i}, c_{2i}$ 는 prototype  $i$ 를 생성한 partition 내에서 값  $V_1, V_2$ 가 발생한 빈도

$R$ 은 거리 계산상 1또는 2를 이용할 수 있으나 여기서는  $R=1$ 을 사용한다. 주어진 예제  $e$ 와 prototype간의 거리는  $\sum_j d(V_1, V_2)$ 로 계산되는데 예제  $e$ 는  $F_j = V_1$ , prototype은  $F_j = V_2$ 를 갖는다고 가정한다. 여기서 추출된 최소 거리를 갖는 prototype의 class가 예제  $e$ 의 class이다.

### 3. 실험적 평가

우리는 PROLEARN 알고리즘을 UCI data repository 중의 4개의 문제를 대상으로 ID3[9] 및 C4.5[10]과 그 효과를 비교하였다. 비교 기준은 정확성과 안정성인데 정확성은 거의 모든 기계 학습 알고리즘이 표방하는 바 일정 숫자의 훈련 data를 통하여 학습한 후 알려지지 않은 test 집합이 들어왔을 때 그 알고리즘이 얼마나 정확하게 그 예제의 class를 판별하는가 하는 것이다.

안정성(stability)은 우리가 제시하는 새로운 알고리즘 판별 기준인데 많은 실제적 기계 학습 응용문제에서 우리는 훈련 data set이 바뀔때 따라 다른 결과가 나오는 것을 경험한다. 좋은 학습 알고리즘은 훈련 data set의 교체에 따른 변화가 적어야 된다고 우리는 믿는다. 또 우리의 PROLEARN 알고리즘은 그 비교 대상인 C4.5나 ID3에 비하여 그것이 이상적 예제를 생성하고 test 예제는 그 prototype과의 거리를 기준으로 class를 판별하므로 더욱 안정성 있는 결과를 나타낼 수 있을 것으로 예측한다.

안정성의 평가 기준은 우리가 훈련 데이터를 바꾸더라도 계속해서 학습의 결과로 남아 있는 것이 몇 % 인가를 본다. 물론 그 %가 높을수록 안정성이 높다.

실험방법은 5-fold cross validation을 30회 사용했으며 예제 집합은 무차별적으로 3개의 집합 - 훈련

data, test data, 안정성 test data(전체의 10%)-으로 나뉜다. 안정성 평가 data는 이것으로부터 추출된 결과가 이론과 비교하도록 한다. 실험의 결과는 <표 1>부터 <표 3>까지 정리되어 있다.

<표 1> 실험문제 설명

<Table 1> Explanation for the Experiment Problems

문 제	예제 수	속성 수	class	비 고
Exclusive-OR	64	6	2	2 무관속성
LED	200	7	10	10% noise
LED-무관속성	500	24	10	10% noise 17 무관속성
Lypography	148	18	4	

<표 2> 정확도 실험결과(%)

<Table 2> Accuracy Result for the Experiments(%)

문 제	PROLEARN	ID3	C4.5
Exclusive-OR	97.09	85.11*	98.81
LED	67.70	58.26*	59.51*
LED-무관속성	67.46	55.45*	67.43
Lypography	81.03	74.69*	78.15*

<표 3> 안정성 실험결과(%)

<Table 3> Stability Result for the Experiments

문 제	PROLEARN	ID3
Exclusive-OR	85.71	39.52*
LED	90.17	63.83*
LED-무관속성	62.76	32.21*
Lypography	79.33	59.56*

표에서 \*로 표시된 것은 PROLEARN이 여타 알고리즘에 비하여 통계적으로 유의하게 우수함을 나타낸 것이다. 표2의 정확도 결과는 PROLEARN이 ID3보다는 우수하고 C4.5와는 비슷하거나 noise 있는 문제에서는 더 우수함을 보여준다. 보다 의미 있는 PROLEARN 알고리즘의 평가는 표3의 안정성 평가다. 예측한대로 PROLEARN은 ID3와 비교할 때 모든 문제에서 훈련 data의 교체가 미치는 영향이 현저히 적음을 보여주었다. 이것은 기계 학습 알고리즘이 실제적 문제에 응용되었을 때 당면하는 noise 및

과소 예제 문제의 한 해결책이라 할 것이다.

#### 4. 결론

이 논문에서 우리는 prototype을 학습하는 것이 가능하며 또 유용하다는 것을 보여주었다. Prototype은 일반 기계 학습 알고리즘이 분류에 필요한 최소한의 속성들로 학습 결과를 나타내는 데 비해 유관하다고 여겨지는 속성들을 중복해 나타냄으로써 정확성을 유지하면서 안정성을 높이는 것을 실험적으로 보여주었다. 이것은 실제 응용 data는 noise가 많고 환경의 변화(예 : 한 병원의 data와 다른 병원의 data가 다름)가 있다는 점을 고려할 때 수치적으로 나타난 결과 이상의 의미가 있다고 할 것이다.

PROLEARN은 아직 그 초기 단계로서 이산적(discrete)인 자료에만 응용가능하나 연속적 data에도 적용 가능할 것으로 믿는다. PROLEARN의 중심 부분인 partitioning 프로세스는 현재로는 하나의 속성만 고려하는데 여러 속성을 동시에 고려하는 학습 알고리즘에 관한 여와 연구들과 연관하여 앞으로 확장되어야 할 부분으로 사료된다.

더불어 이 논문에서는 다루지 않았으나 prototype 학습은 단순히 분류뿐만 아니라 추론(inference)에도 적용 가능하다고 우리는 믿는다. 추론의 과정은 학습이 주어진 예제의 속성과 값을 바탕으로 class를 추정하는 데 반해, 부분적인 속성-값 집합과 class를 주면 나머지 속성의 값을 추정하는 과정을 말하며 이것이 가능하다면 문제 해결 시스템(Problem Solving System) 등에 획기적인 효과를 가져올 것이다.

#### ※ 참고문헌

- [1] Aha, D., Kibler, D. and Albert M.. Instance based learning algorithms. *Machine learning*, volume 6, pp 37-66. Boston, MA:Kluwer Publishers, 1991.
- [2] Aydin, A. and Pearce, J.. Prototype effects in categorization by pigeons. *Journal of Experimental Psychology:Animal Behavior Processes*, volume 20(n3) pp.264-277, 1994.

- [3] Clark, P. and Boswell, R. Rule Induction with Cn2: some recent improvements. *European Working Session on Learning*, 1991.
- [4] Cost, S. and Salzberg, S.. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, volume 10, pp. 57-78. Boston, MA: Kluwer Publishers, 1993
- [5] Duda, R., and Hart P.. *Pattern classification and scene analysis*. New York: John Wiley & Sons, 1973.
- [6] Huttenlocher, J., Hedges, L. and Duncan, S.. Categories and particulars: prototype effects in estimation spatial location. *Psychological Review*, volume 98(n3) pp.352-376, 1991.
- [7] Medin, D., Altom, M. and Murphy, T.. Given versus induced category representations: Use of prototype and exemplar information in classification. *Journal of Experimental Psychology: Learning, Memory, Cognition*. volume 10(n3) pp.333-352, 1984.
- [8] Murphy, P. and Aha, D.. UCI repository of machine learning databases [machine readable data repository]. Tech. Rep., University of California, Irvine, 1994.
- [9] Quinlan, J. R.. Induction of decision trees. *Machine Learning*, volume 1. Boston, MA: Kluwer Publishers, 1986.
- [10] Quinlan, J. R.. C4.5: *Programs for Machine Learning*, Morgan Kaufmann, San Mateo. CA., 1993.
- [11] R. Rosenblatt, J.. Modeling by shortest data description. *Automatica*, volume 14 pp. 465-471, 1958.
- [12] Rissanen, F.. The perception: a probabilistic model for information storage and organization in the vrain. *Psychological Review*, volume 65(n6) pp.386-408, 1978.
- [13] Skalak, D.. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Proceedings of the Eleventh International Conference on Machine Learning*, New Brunswick, NJ. Morgan Kaufmann, 1994.
- [14] Zhang, J.. Selecting typical instances in instance-based learning. In *em Proceedings of the Ninth International Conference on Machine Learning*, New Brunswick, NJ. Morgan Kaufmann, 1992..

송 두 현



1981 서울대학교 계산통계학과 졸업 (B.S.)  
 1983 KAIST 전산학과 졸업 (M.S.)  
 1995 University of California at Irvine 전산학과(박사 수료)  
 1983 ~ 1986 KAIST 연구부 연구원  
 1997 ~ 현재 용인송담대학 컴퓨터소프트웨어과 교수  
 관심분야 : 인공지능, 데이터베이스, XML