변형된 Category Utility를 이용한 점진 개념학습

Incremental Conceptual Clustering Using Modified Category Utility

김표재*, 최진영**
서울대학교 전기컴퓨터공학부

Pyo Jae Kim* and Jin Young Choi**
School of Electrical Engineering and Computer Science
Seoul National University
E-mail : *pjkim@neuro.snu.ac.kr, **jychoi@neuro.snu.ac.kr

ABSTRACT

점진적 개념 학습 알고리즘인 COBWEB은 클래스 정보가 주어지지 않은 사례들(instances)을 분류하기 위하여 사례의 속성과 값에 근거하여 학습하며 각 노드가 유사한 사례들의 집합인 클래스에 해당하는 분류 트리를 생성하는 알고리즘이다. 유사한 사례들을 같은 클래스로 분류하기 위한 기준으로 category utility가 사용되며 이는 클래스 내부의 유사도와 클래스 간의 차이점을 최대화 하는 방향으로 클래스를 분류한다. 기존의 COBWEB에 사용되는 category utility는 클래스 사이즈와 예측 정확성 사이의 tradeoff 관계로 볼 수 있으며, 이로 인하여 예측 정확성은 약간 감소하나 클래스 사이즈가 커지는 방향으로 학습이 진행 될 수 있는 편향성(bias)를 가지고 있다. 이는 분류 트리에 불필요한 클래스 노드들(spurious nodes)을 생성하게 하여 학습 결과인 클래스 개념을 이해하는데 어렵게 한다. 본 논문에서는 클래스와 그에 속하는 사례들의 속성-값 분포를 고려하여 클래스와 속성의 연관성에 비례한 가중치를 더한 변형된 category utility를 제안하고, dataset에 대한 실험을 통하여 제안된 category utility가 기존의 큰 클래스 사이즈를 선호하는 bias를 완화 시킴을 보이고자 한다.

Key words : incremental conceptual clustering, COBWEB, category utility

I. 서 론

인간은 경험이나 관찰을 통하여 획득한 정보를 근거로 이를 요약하거나 조직화 과정을 통하여 개념(concepts)들을 학습하며 이를 다음 판단의 기준으로 사용한다. 이러한 인간의 학습 과정은 다음과 같은 몇몇 특징들을 가지고 있다. 학습에 사용되는 다양한 정보들은 그들 사이의 관계가 명확히 주어 지지 않을 수 있으며, 그들이 어떠한 개념을 표현하는 지가 명시적으로 나타나지 않을 수 있다. 또한 다양한 정보들을 통해 습득되는 개념의 개수도 학습 방법에 따라 달라 질 수 있다. 또한 새로운 사례에 대하여 점진적으로 학습을 반복하며 이를 통하여 기존의 개념을 수정하거나 새로운 개념을 생성해 나간다.

이러한 인간의 학습능력을 모방하기 위하여 다양한 기계학습 방법들이 연구되어 왔으며 Michalski & Stepp[1]은 이를 개념 분류화 문제(conceptual clustering)로 정의 하였다. 개념 분류 문제들은 개념의 계층적인 조직화(concept hierarchy), 무감독 학습(unsupervised learning), 점진적 학습(incremental learning)등과 같은 공통된 특징을 가지며 EPAM[2], UNIMEM[3], COBWEB[4]과 같은 다양한 학습 알고리즘이 제안 되었다.

Fisher에 의해 제안된 점진적 개념 학습인 COBWEB은 사례들의 발생 빈도 확률을 토대로 형성되는 클래스를 노드로 가지는 분류 트리(classification tree)를 학습하며, category

utility[5]를 분류기준으로 사용하여 새로운 사례들을 이전 내용의 재 학습 없이 점진적으로 학습할 수 있는 알고리즘이다. COBWEB은 비교적 잘 정의된 분류 함수와 양방향 학습 연산자들을 이용하여 점진학습의 문제점인 사례의 순서의 영향을 어느 정도 해결하면서 안정적인 성능을 가진다. 그러나 분류 기준으로 사용되는 category utility의 편향성이 존재하며, 이로 인하여 분류 트리에 불필요한 중간 노드들이 생성되는 문제점을 가지고 있다.

본 논문에서는 COBWEB에 사용되는 category utility의 편향성을 완화시키기 위해 변형된 식을 제안하고자 한다. 동일 클래스에 속하는 속성-값들의 분포를 고려하여 클래스와 속성의 연관성에 비례하는 가중치를 더한 변형된 category utility를 제안하고 이를 여러 dataset의 분류에 적용하여보고 기존의 식을 이용한 실험 결과와의 비교를 통하여 편향성이 완화됨을 확인하고자 한다.

II. 본 론

2 COBWEB

Fisher에 의해 제안된 COBWEB[4]은 클래스 정보가 주어지지 않은 사례들(instances)을 이를 구성하는 속성들과 속성이 가질 수 있는 값의 분포에 근거하여 유사한 클래스로 분리하는 학습 알고리즘이다. 학습 결과는 분류 트리 형태로 나타나며, 학습에 사용되는 모든 사례의 정보는 분류 트리의 루트 노트에 발생 빈도 확률로 저장된다. 분류 트리의 각 노드들은 유사한 사례들이 모인 클래스를 나타내며 부모 노드는 자식 노드 보다 좀더 일반적인 개념을 나타낸다.

속성과 값의 쌍으로 표현된 사례들은 category utility라는 분류 함수에 따라 루트 노드에서부터 가장 유사한 노드(클래스)로 분류되고 동일한 과정을 단말 노드에 이르기 까지 반복하게 된다. 이렇게 형성된 분류 트리의 각 노드에는 같은 클래스에 속하는 사례들의 모든 속성과 값의 발생 빈도를 확률로서 저장하며 이 확률 표현을 기반으로 노드가 나타내는 개념을 정의 할 수 있다.

COBWEB은 새로운 사례들의 학습과 분류가 동시에 일어나며 category utility에 근거하여 기존의 내용의 재학습 없이 분류 트리 상의 한 노드에 위치하거나 새로운 노드를 생성한다. 이때 사용되는 학습 연산자를 Incorporate와 Create new disjunct라 한다. Incorporate의 경우 새로 입력되는 사례를 기존의 노드의 하위 클래스에 포함시키는 연산자이며, Create new disjunct는 새로 입력된 사례가 기존 분류 클래스 노드들과 유사성이 적을 때 새로운 클래스 노드를 생성하는 연산자이다.

이렇게 새로운 사례들을 점진적으로 학습 하는 방법은 사례의 주어지는 순서에 따라 다른 결과를 나타낼 수 있다. 순서에 의하여 분류 트리가 최적화 되지 못하는 문제를 해결하기 COBWEB에서는 Merge와 Split라는 학습 연산자를 추가하였다. Merge 연산자는 분류 각 단계에서 2개의 최적 노드(현재 노드의 자식 노드들 중 category utility가 높은 2개의 노드들에 해당함)의 병합을 고려한다. 만약 병합으로 생성된 새로운 분류 트리의 category utility가 기존의 것보다 크다면 두 노드는 병합되어 단일 노드(클래스)를 형성한다. 이와 반대되는 연산인 Split는 노드의 분할을 일으킨다. 최적의 노드의 자식 노드들을 한 계층 위로 끌어 올림으로써 분류 품질의 향상을 도모한다. 그림 1은 Merge와 Split 연산자를 나타낸다.

추가 학습 연산자의 사용에도 불구하고 사례의 순서에 의한 영향은 존재하며 이를 해결하기 위해 Biswas & Fisher는 분류 품질이 최적화 되도록 사례의 순서를 조절하여 학습할 수 있는 ITERATE[6]를 제안하였다.

COBWEB은 확률적인 개념 표현으로 인하여 속성 값들이 명목 값(nominal value)만을 허용하는 제약 조건이 있었으나 Gennari & Fisher에 의하여 수치적 속성을 허용하는 CLASSIT[7]로 확장되었다.

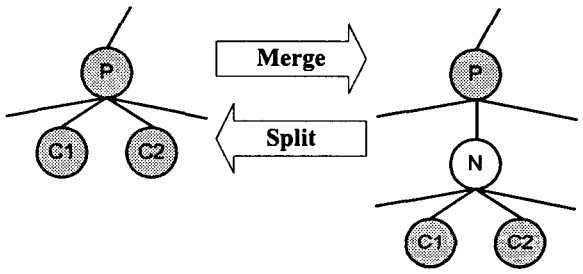

그림1. COBWEB의 Merge, Split 연산자

2.1 Category Utility와 bias

Fisher는 사례를 분류하기 위한 기준으로써 Gluck & Corter[5]에 의해 제안된 category utility를 일반화하여 사용하였다. 확률 부합 전략(*probability matching strategy*)에 기반한 category utility는 클래스 내부의 유사도(intraclass similarity)와 클래스 간 차이점(interclass dissimilarity)의 tradeoff로 정의된다. 임의의 사례 집합에 대하여 상호 배타적인 자식 노드들을 가지는 부모 노드의 category utility는 다음과 같다.

$$CU = \frac{1}{n}\sum_{k}^{n} P(C_k)\sum_{i}\sum_{j}[P(A_i = V_{ij} \mid C_k)^2 - P(A_i = V_{ij})^2] \quad (1)$$

여기서 CU는 분류 트리상의 임의의 노드의 category utility를 의미하며 n 은 자식 노드(클래스) C_k 의 개수를 나타낸다. 조건부 확률 $P(A_i=V_{ij}|C_k)$ 는 클래스 정보를 고려 하였을 때, 속성 A_i가 값 V_{ij}를 가질 확률을 의미한다. 식(1)에 따르면 category utility는 클래스 정보가 주어졌을 경우 그렇지 않았을 때보다 정확히 예측되는 속성 값들의 개수의 증가량의 추정 값을 의미한다.

비슷한 관점에서 category utility는 클래스 사이즈 $P(C_k)$ 와 클래스의 속성 값들에 대한 예측 정확성 $\sum_i\sum_j[P(A_i=V_{ij}|C_k)^2-P(A_i=V_{ij})^2]$ 간의 tradeoff로 해석할 수 있다. 클래스 사이즈에 비례하는 $P(C_k)$가 포함됨으로써 category utility는 좀 더 큰 사이즈의 클래스를 선호하는 bias를 가지게 된다. 이러한 bias는 비슷한 유형의 사례가 연속해서 제시되는 경우에 분류 트리에 생성되는 클래스들을 왜곡시킬 가능성을 가지고 있다. 특히 큰 클래스를 선호하는 bias는 학습 초기에 큰 영향을 끼친다. 이는 학습 초기에는 새로운 사례로 인한 값의 변화가 예측 정확성 값 보다 클래스 사이즈 값이 더 크기 때문이다. 즉 새로 입력되는 사례가 기존의 학습된 사례들과 완전히 다른 사례가 아니라면 분류 트리는 이를 한 클래스에 속하는 사례로 분류할 가능성이 크게 된다.

이러한 bias는 Merge와 Split의 학습 연산자로 그 효과가 어느 정도 완화가 되지만 분류 트리에 의미가 모호한 중간 노드들이 생성되는 것을 완전히 방지할 수 없다. 이러한 의미 없는 노드들(spurious nodes)은 분류 트리의 클래스로 형성되는 개념들을 이해하기 힘들게 한다. 이러한 bias문제를 극복하기 위하여 본 논문에서는 변형된 category utility함수를 제안하고자 한다.

2. 2 Modified category utility

Category utility의 bias의 영향은 클래스 사이즈와 예측 정확성 사이의 tradeoff로 인해 발생된다. 학습 초기에 어느 정도의 예측 정확성 감소를 감수하면서 보다 큰 클래스를 구성하도록 사례를 분류하면 category utility에 기반한 분류 품질은 높아지게 된다. 이는 분류 트리에 상대적으로 적은 수의 공통된 속성-값을 가지는 사례들이 모인 일반적이고 사이즈가 큰 클래스가 생성됨을 의미한다. 이러한 노드(클래스)는 같은 레벨의 다른 노드에 비해 개념을 이해하기 어렵거나 다른 노드들에서 배제된 사례들의 집합이라는 단순한 의미를 가지는 경우가 대부분이다. 또한 이러한 불필요한 노드(spurious nodes)의 생성은 분류 트리의 깊이를 깊어지게 한다.

Zoo dataset[8]에 대한 기존의 category utility를 사용한 경우의 학습 결과는 그림 2와 같다. 루트 노드 바로 밑에 포유류를 제외한 다른 모든 사례들을 포함하는 노드가 발생함을 확인할 수 있다.

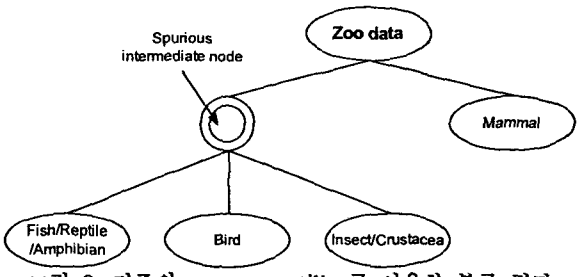

그림 2. 기존의 category utility를 이용한 분류 결과 (zoo)

앞에서 설명한 bias의 영향을 줄이기 위하여 다음과 같은 변형된 category utility를 제안한다.

$$CU=\frac{1}{n}\sum_k^n P(C_k)\sum_i\sum_j[(P(A_i=V_{ij}|C_k)-\frac{1}{N_j}) \quad (2)$$
$$\{P(A_i=V_{ij}|C_k)^2-P(A_i=V_{ij})^2\}]$$

N_j 는 속성 A_i가 가질 수 있는 값 V_{ij} 의 경우의 수이다.

위 식에는 기존의 category utility에 한 클래스에 속한 사례들의 속성-값의 분포를 고려하는 가중치 항이 포함되었다. 즉 속성 A_i가 클래스 내부에서 V_{ij} 인 확률이 V_{ij}에 관계없이 고른 분포를 보인다면 속성 A_i 는 클래스의 category utility를 결정하는데 영향을 적게 미치게 된다. 이것은 클래스와 속성의 연관성을 고려함으로써 클래스 내부의 유사도가 큰 방향으로 학습이 진행되게 하는 효과가 있다. 이로 인하여 클래스 사이즈에 대한 bias가 완화되는 효과를 얻고자 한다.

3. 실험 결과

제안된 category utility 함수의 성능 검증을 위하여 모의 시험을 실시하였다. Zoo dataset와 Soybean diseases dataset[8]에 대하여 기존의 category utility와 변형된 식을 이용하는 COBWEB을 적용하여 결과를 비교하였다.

Zoo dataset는 16개의 속성과 각 사례들이 속한 클래스 정보로 이루어져있다. 클래스 정보를 무시하고 16개의 속성만을 고려하여 분류하고 이를 미리 주어진 클래스 정보와 비교

하여 분류 품질을 확인하여 보았다. 두 경우 모두 오차율을 4%정도이며 생성되는 분류 트리는 그림 2와 3과 같다.

Soybean diseases dataset은 35개의 속성으로 구성되며 이들 속성에 따라 4가지의 질병으로 분류되는 soybean data들을 가지고 있다. 기존의 category utility와 제안된 식을 이용한 결과는 각각 그림 3, 4에 해당한다.

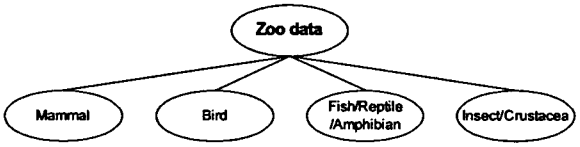

그림 3. 변형된 category utility를 이용한 분류 결과 (zoo)

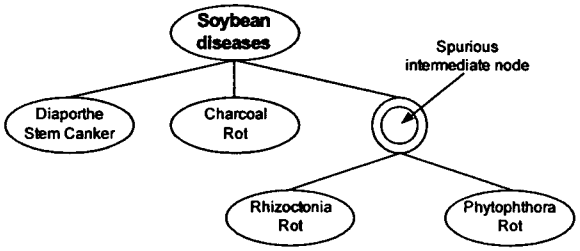

그림 4. 기존의 category utility를 이용한 분류 결과 (soybean)

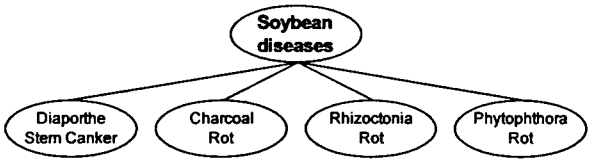

그림 5. 변형된 category utility를 이용한 분류 결과 (soybean)

그림 2에서 5까지의 결과에서 확인 할 수 있듯이 제안된 category utility는 오차율의 변화 없이 불필요한 노드들의 생성을 억제하는 효과를 가짐을 확인할 수 있었다. 이러한 효과는 분류 트리를 좀더 간결하게 하며, 클래스 내부의 유사도가 증가하기 때문에 클래스의 개념을 이해하기 쉽게 한다. 또한 분류 트리에 의미 없는 노드들이 생성이 되지 않아 양방향 연산자의 적용 회수도 감소시킨다. (연산자 적용 회수는 사례의 순서에 민감한 결과를 보이므로 구체적인 수치는 생략하였다.)

III. 결 론

본 논문에서는 COBWEB의 category utility의 bias로 인한 불필요한 노드의 생성 현상을 개선하기 위하여 변형된 category utility를 제안하였다. 제안된 category utility는 클래스와 속성의 연관성 대한 가중치 항을 고려하여, 예측 정확성을 높이는 동시에 큰 클래스 사이즈를 선택하는 bias의 영향을 완화 시킨다. Zoo와 soybean dataset을 이용한 실험 결과에서 제안된 category utility는 분류 품질에 영향 없이 불필요한 노드를 생성을 억제하고 좀더 명확한 형태의 분류 트리를 생성함을 확인 할 수 있었다.

향후 과제로는 제안된 category utility가 수치적 속성에 대한 가충치 항을 고려할 수 있도록 확장하고 다양한 dataset에 대한 실험을 통하여 이의 효과를 확인하고자 한다. 또한 실험을 통하여 제안된 category utility의 의미를 분석하고 좀 더 개선된 식을 제안하고자 한다.

감사의 글: 본 연구는 산업자원부 차세대 신기술 개발사업(슈퍼 지능칩 및 응용기술 개발 과제)의 지원을 받아 수행되었습니다.

IV. 참고문헌

[1] R. Michalski and R. E. Stepp, "Learning from observation: Conceptual clustering", *Machine Learning: An Artificial Intelligence Approach*, R. Michalski, J. Carbonell, and T. Mitchell, Eds. Palo Alto, CA: Tioga Press, 1983, pp. 331 -364

[2] E. A. Freigenbaum and H. Simon, "EPAM-like models of recognition and learning", *Cognitive Science*, Volume: 8, 1984, pp. 305 -336

[3] M. Lebowitz, "Experiment with incremental concept formation: UNIMEM", Machine Learning, Volume: 2, 1987, pp. 103 -138

[4] D. Fisher, "Knowledge acquisition via incremental conceptual clustering", Machine Learning, Volume: 2 Issue: 2, 1987, pp. 139 -172

[5] M. Gluck and J. Corter, "Information, uncertainty, and the utility of categories", in Proc. 7th Ann. Conf. Cognitive Sci. Soc., Irvine, CA, 1985, pp. 283 -287

[6] G. Biswas, J. Weinberg, D. Fisher, "ITERATE: A Conceptual Clustering Algorithm for Data Mining", IEEE Trans. System Man and Cybernetics, Volume: 28 Issue: 2, 1998, pp. 219 -230

[7] J. Gennari, P. Langley, D. Fisher, "Models of incremental concept formation", *Artificial Intelligence*, Volume: 3, 1989, pp. 11 -61
[8] University of California, Irvine, Information and Computing Science FTP (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/)