점진적 개념학습의 클러스터 응집도 개선

백 혜 정† · 박 영 택††

요 약

요즘, 인터넷 등장 이후 폭발적으로 증대되는 웹 정보를 효율적으로 사용하기 위한 시스템들이 요구되고 있다. 이러한 요구를 해결하기 위해 개발된 시스템들은 서비스 정보의 질을 향상시키기 위하여 클러스터링 기법을 이용하고 있다. 클러스터링은 무질서한 데이터들의 상호 연관관계를 정의하고 이를 통하여 보다 체계적으로 데이터를 군집화하는 것이다. 클러스터링을 이용한 시스템은 비슷한 내용을 묶어 사용자에게 제공함으로, 사용자는 보다 효율적으로 정보를 파악할 수 있다. 그래서 이전 연구에서 대량의 데이터를 효율적으로 클러스터링 하기 위하여 통합 클러스터링 방식을 제안하였다. 이 방식은 COBWEB 알고리즘을 이용하여 초기 클러스터를 생성한 후, Etzioni 알고리즘을 이용하여 클러스터링을 생성하는 방식이다. 본 논문은 이러한 기존의 통합 클러스터링 방식의 정확성과 효율성을 높이기 위하여, 다음 두 가지 방식을 제안한다. 첫째, 클러스터할 데이터의 속성의 가중치를 고려한 클러스터링 방식을 제안한다. 둘째, 기존의 클러스터링 방식의 효율성을 지원하기 위하여, 초기 클러스터를 생성하는 평가 함수를 재정의한다. 본 논문에서 제안하는 클러스터링 방식은 방대한 양의 데이터를 효율적으로 처리 할 수 있으며 데이터의 입력 순서의 의존도를 줄여, 데이터를 효과적으로 클러스터, 양질의 사용자 프로파일 구축에 도움을 주게 된다.

The Study on Improvement of Cohesion of Clustering in Incremental Concept Learning

Hey-Jung Baek† · Young-Tack Park††

ABSTRACT

Nowdays, with the explosive growth of the web information, web users increase requests of systems which collect and analyze web pages that are relevant. The systems which were develop to solve the request were used clustering methods to improve the quality of information. Clustering is defining inter relationship of unordered data and grouping data systematically. The systems using clustering provide the grouped information to the users. So, they understand the information efficiently. We proposed a hybrid clustering method to cluster a large quantity of data efficiently. By that method, We generate initial clusters using COBWEB Algorithm and refine them using Ezioni Algorithm. This paper adds two ideas in prior hybrid clustering method to increment accuracy and efficiency of clusters. Firstly, we propose the clustering method considering weight of attributes of data. Second, we redefine evaluation functions which generate initial clusters to increase efficiency in clustering. Clustering method proposed in this paper processes a large quantity of data and diminish of dependancy on sequence of input of data. So the clusters are useful to make user profiles in high quality. Ultimately, we will show that the proposed clustering method outperforms the pervious clustering method in the aspect of precision and execution speed.

키워드 : 클러스터링(Hybrid Clustering), 가중치(Weight), 평가 함수(Evaluation Function), COBWEB, Etzioni

1. 서 론

인터넷의 등장 이후 폭발적으로 증가하는 웹 정보를 효율적으로 사용하기 위하여 웹 에이전트가 개발되고 있다. 웹 에이전트는 사용자가 인터넷 정보 행위를 모니터하여 사용자의 관심 정보를 학습하고 사용자가 필요로 하는 웹 상의 정보를 자동 제공하는 지능형 시스템이다. 웹 에이전트의 성능은 사용자의 관심을 파악에 좌우되는데, 웹 에이전트는 사용자의 관심을 파악하기 위한 방법으로 클러스터링을 이용한다[7]. 이는 클러스터링 기법이 무질서한 데이터들의 상호 연관관계를 정의하고 이를 통하여 보다 체계적으로 데이터들을 군집화하며, 각 군집된 데이터의 속성을 파악할 수 있기 때문이다[1].

클러스터링 기법에는 점진적 클러스터링 방식과 일괄처리 클러스터링 방식이 있다. 점진적인 클러스터링 방식은 입력데이터들을 하나씩 입력받아 유사한 데이터들을 클러스터링하는 방식이며, 일괄처리 클러스터링 방식은 입력데이터들을 모두 입력받아 한번에 처리하는 방식이다. 점진적 클러스터링 방식은 입력 데이터를 하나씩 처리하면서 유사

※ 본 논문은 숭실대 교내 연구비 지원으로 이루어 졌음.
† 준 회 원 : 숭실대학교 대학원 컴퓨터학과
†† 정 회 원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 2002년 2월 2일, 심사완료 : 2003년 6월 12일

한 입력 데이터들을 클러스터링 함으로, 일괄처리 클러스터링 방식보다 효율적이며, 방대한 데이터를 클러스터링 하는데 적합하다. 하지만 점진적 클러스터링 방식은 데이터의 입력 순서에 따라서 생성된 클러스터의 내용이 정확하지 못하다는 단점이 있다. 본 논문에서는 점진적인 방식의 속도 효율성을 유지하면서, 입력 순서 의존적인 문제를 완화시키고 클러스터내의 응집도를 높이기 위한 방식을 구현하는 것을 목적으로 한다.

본 연구에 앞선 연구에서 점진적인 방식의 속도 효율성을 유지하면서, 입력 순서 의존적인 문제를 완화시키기 위해서 통합 클러스터링 방식을 제안하였다. 통합 클러스터링 방식은 COBWEB 알고리즘을 이용하여 각 데이터간의 상호 연관성을 정의하고, 상호 연관성을 기준으로 초기 클러스터를 생성한다. 초기 클러스터들은 Etzioni 알고리즘을 이용하여 최종 클러스터링을 생성하는 방식이다. 본 연구에서는 입력 의존적인 문제를 효율적으로 완화시키고, 클러스터내의 응집도를 최대화하기 위하여 COBWEB과 Etizioni의 평가 함수(Evaluation Function)를 수정 제안하여 각 속성에 가중치를 주는 방안을 제안한다. 그리고, 통합 클러스터링 방식에서 속도 효율성을 유지하면서, 입력 순서 의존적인 문제를 최소화하는 초기 클러스터를 생성하는 방식을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 고전적인 클러스터링 알고리즘방식을 설명하고, 3장에서는 본 논문에서 제안하는 클러스터링 방식의 시스템 구조를 설명한다. 4장, 5장에서는 본 논문에서 제안하는 가중치 방식과 초기 클러스터를 효율적으로 선택하기 위해 제안된 평가함수를 자세히 설명하도록 한다. 6장에서는 쇼핑몰 데이터를 대상으로 실험을 하여 본 논문에서 제안하는 클러스터링에서의 확장성과 효용성을 기술한다. 끝으로 7장에서 결론 및 향후연구 방향에 대해서 기술한다.

2. 관련 연구

클러스터링 알고리즘은 크게 점진적 클러스터와 비 점진적 클러스터로 나뉘어 진다. 점진적 클러스터는 입력되는 순서에 따라 입력 데이터들을 하나씩 처리하면서 유사한 입력 데이터들을 클러스터링 하는 방식을 말한다. 이와는 대조적으로 비 점진적 클러스터는 입력 데이터들을 한꺼번에 입력받고 클러스터에 대한 계산을 마친 뒤 유사한 입력 데이터들을 클러스터링 하는 방식을 말한다. 본 절에서는 클러스터링에서 많이 사용되고 있는 클러스터인 COBWEB 방식과 K-MEAN 방식을 설명한다. COBWEB 방식은 점진적인 클러스터로서 각 데이터의 상관관계가 계층적인 구조로 구성된다. 이 방식은 최종 클러스터를 정하기 위해서 Threshhold값을 정할 필요가 있다. 이에 반해 K-MEAN 방식은 일괄처리 방식으로서 사용자가 클러스터 하기 전 클러스터 수를 미리 정하여 클러스터링 하는 방식이다[3, 4]. 다음은 본 논문에 대한 관련연구로써 이에 대하여 설명한다.

2.1 COBWEB

Cobweb은 원래 인간의 점진적인 개념학습을 모델링 하기 위해 개발되었다. 인간과 마찬가지로 Cobweb은 관찰을 통해서 개념을 형성해 가고 형성된 개념을 이용하여 새로운 예제를 분류할 수 있다[8]. Cobweb은 개념적 클러스터링(Conceptual Clustering) 알고리즘으로 다른 기계학습과 비교해서 다음과 같은 특성을 가진다. 첫째, 분류 대상이 되는 개념을 계층적인 구조로 구성한다. 둘째 하향식 분류(Top-down Classification)를 수행한다. 셋째 비감독 학습(Unsupervised Learning)방식이다. 넷째 점진적인 학습(Incremental Learning)을 수행한다. 다섯째 새로운 학습대상에 대하여 힐 클라이밍(Hill Climbing) 기법을 사용하여 효과적인 해를 구하는 방식으로 수행된다.

Cobweb은 새로운 문서를 입력으로 받아들이면 트리의 각 레벨에서 어떤 학습 연산자를 적용할 것인지를 평가함수의 값에 따라서 적용한다. 적용된 학습 연산자에 의해서 입력된 문서는 자신이 속할 클러스터를 찾아가게 된다. Cobweb의 학습 연산자는 Incorporate, Create-new-disjunct, Merge, Split의 4가지 연산자로 이루어져 있다. 트리의 각 노드에는 평가함수의 값을 계산하기 위해서 노드의 상태를 저장하고 있다. 그리고 학습 연산자를 적용하여 트리의 구조가 변할 경우에는 노드의 상태 정보도 변경된다.

2.2 K-MEANS

K-means 알고리즘은 매우 간단하면서도 효과적인 알고리즘이다. 여기서 의미하는 K는 몇 개의 클러스터를 생성할 것인지를 결정하는 초기 값이다. 즉 사용자는 해당하는 데이터 집합에 대하여 몇 개의 클러스터를 생성하는 것이 적당할 것인가를 미리 생각하고 K개의 초기값을 설정하게 되며 실행된 결과를 분석하여 더 좋은 결과를 가지도록 초기값 K를 다시 수정할 수 있다.

K-means 알고리즘이 간단하면서도 효율적인 이유는 모든 데이터들을 벡터 공간상에 설정하게되며 부여된 K개 만큼의 클러스터를 위해 단순한 거리 계산을 이용한 중심값을 찾아가는 알고리즘 방식을 적용하기 때문이다. 다음은 K-means 알고리즘은 다음과 같다. 첫째, 클러스터의 수(K개)를 선택하고 클러스터의 중심점(center)들을 초기화한다. 둘째, 모든 입력 클러스터의 중심점과 입력 데이터(벡터)들에 대하여 거리를 비교한다. 이때 가장 작은 거리를 가지는 클러스터 중심점에 대해서 해당 데이터는 그 클러스터에 속하게 된다. 세째, 클러스터의 중심점을 다시 계산한다. 계산된 클러스터 중심점에 대하여 각 데이터들의 거리를 다시 계산하고 작은 거리를 가지는 클러스터로 데이터를 재

분배 한다. 넷째, 더 이상의 클러스터 중심점의 변화가 없으면 수행이 끝나게 된다. 이처럼, 초기 클러스터의 중심값과 클러스터 형태가 시간이 지남에 따라 변화 된다[6, 10].

3. 시스템 구조

클러스터링은 무질서한 데이터들의 상호 연관관계를 정의하고 이를 통하여 보다 체계적으로 데이터들을 군집화하는 것이다. 현재 클러스터링 기법은 웹 문서의 증가와 사용자의 서비스 질 향상을 위하여 많이 이용되고 있다. 많은 데이터를 보다 빠르고 효율적으로 처리하기위해서, 점진적 클러스터링 방식을 이용한다. 점진적인 클러스터링 방식은 많은 양의 데이터분류에 효과적인 특성을 가지고 있는 반면에, 입력 순서에 따라 생성되는 클러스터가 다르다. 본 논문에서는 이러한 점진적인 방법의 효율성을 최대화 하면서, 입력 의존적인 방법을 최소화하기 위해서 기존에 제안된 통합된 방식의 평가함수를 재정의 한다.

기존의 통합된 방식은 Top-down 방식이며 개념적 클러스터링(Conceptual Clustering) 방식인 Cobweb을 이용하며, Bottom-up 클러스터링 방식인 Etzioni의 교차기반 클러스터링(Intersection-based) 방식을 이용한다. COBWEB은 속도 면에서는 빠른 클러스터링을 수행하지만 입력 문서의 순서에 종속적이며 클러스터링을 위한 분류 후처리를 필요로 한다. Etzioni의 클러스터링은 입력 데이터의 순서에 비종속적이며, 클러스터의 수가 자동으로 정해지는 클러스터링를 수행하는 반면에 많은 계산량을 필요로 해서 실행 속도면에서 취약점을 가진다. 그래서 두가지 방식을 통합함으로 두 방식의 장점만을 취하였다. 이 때, 통합 클러스터링 방식을 효과적으로 통합하기 위해서는 COBWEB이 생성하는 분류 트리로부터 효율적인 초기 클러스터 생성이 중요하다. 본 논문에서는 효율적인 초기 클러스터 생성을 위한 평가함수를 제안하여, 클러스터링의 효율성을 높인다. 또한 본 논문에서는 클러스터할 데이터의 속성의 중요도를 고려한 방식[11]을 제안한다.

다음 그림은 본 논문에서 제안하고 있는 클러스터링 방식 구조도이다.

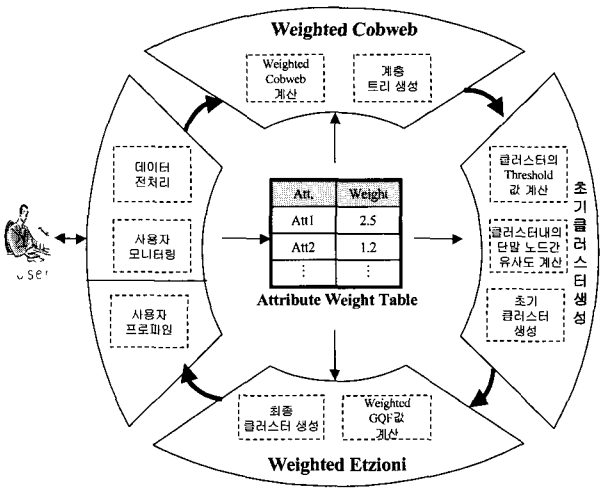
이때, 가중치를 고려하는 통합 클러스터링의 방식의 절차는 다음과 같다.

[1 단계] Weighted COBWEB 클러스터링을 적용하여 계층적 구조를 생성

[2 단계] 초기 클러스터 생성 평가함수를 이용하여 초기 클러스터를 생성

[3 단계] Weighted Etzioni을 이용하여 초기 클러스터 병합통한 최종 클러스터를 생성

이와 같은 단계를 통하여 최종 클러스터를 생성하게 된다. 본 논문에서 제안하는 가중치 기반의 클러스터링은 통합 클러스터링 방식에 적용되는 COBWEB 클러스터링과 Etzioni 클러스터링의 평가 함수를 재정의 한 것으로 4장에 설명한다. 입력 의존적인 문제를 최소화하고, 클러스터의 정확도를 높이기 위한 초기 클러스터링 생성을 위한 방법은 5장에서 제안한다.



4. 가중치 기반의 클러스터링

본 논문에서는 클러스터할 속성에 가중치를 고려하여 클러스터링을 수행 할 수 있는 가중치 기반 통합 방식의 클러스터링을 제안하였다. 4.1절에서는 초기 클러스터를 생성하기 위하여 제안된 Weighted COBWEB 알고리즘을 설명하고, 4.2절에서는 최종 클러스터를 생성하기 위하여 제안된 Weight Etzioni 알고리즘을 설명한다.

4.1 Weighted COBWEB 클러스터링

COBWEB은 원래 인간의 점진적인 개념학습을 모델링 하기 위해 개발되었다. 인간과 마찬가지로 COBWEB은 관찰을 통해서 개념을 형성해 가고 형성된 개념을 이용하여 새로운 예제를 분류할 수 있다. COBWEB은 데이터들의 상호 연관성을 이용하여 계층적 구조의 클러스터링 트리를 생성한다. COBWEB은 계층적 구조를 생성값으로 Category Utility[5]를 기반으로 하는 평가 함수를 사용한다. 이때, 평가 함수는 Predictiveness와 Predictability값을 고려한 것이다. Predictiveness는 한 데이터의 속성의 값이 주어졌을 때, 특정 클러스터에 속할 확률을 의미하며, Predictability는 한 데이터가 특정 클러스터일 때, 속성이 특정 값을 가질 확률을 의미한다. 이를 식으로 나타내면 식 (1)과 같다.

$$\sum_k \sum_i \sum_j P(C_k|A_i = V_{ij}) P(A_i = V_{ij}|C_k) \qquad (1)$$

위에서 $P(C_k)$란 클러스터 k의 전체에 대한 비율이고,

$P(A_i = V_{ij} | C_k)$는 주어진 클러스터에 대하여 개체의 속성(A_i)이 특정 값(V_{ij})을 가지는 확률을 나타내며, i는 속성의 개수, j는 학습개체의 개수를 나타낸다. 이때, 각 속성의 값을 $P(A_i = V_{ij})$ 와 각 속성의 가중치($W(A_i)$)라 했을 때, 이를 식 (1)에 적용하면 다음과 같은 식을 얻을 수 있다.

$$\sum_k \sum_i W(A_i) \sum_j P(A_i = V_{ij}) P(C_k | A_i = V_{ij}) \frac{P(A_i = V_{ij} | C_k)}{} \quad (2)$$

식 (2)를 베이지언 정리 ($P(A_i = V_{ij}) P(C_k | A_i = V_{ij}) = P(C_k) P(A_i = V_{ij} | C_k)$)를 이용하면 가중치를 고려한 값은 다음과 같이 정리할 수 있다.

$$\sum_k P(C_K) \sum_i W_i \sum_j P(A_i = V_{ij} | C_k)^2 \quad (3)$$

Gluck and Corter에 의한 Category Utility는 식 (1)에서 계산한 어떤 개체의 속성이 기대되는 기대치의 합 식 (3)에서 분류를 고려하지 않은 기대치값 식 (4)를 뺀값을 부류의 개수(K)로 나눈 값으로 정의했다[5]. 이때, 속성이 가중치를 가진다고 하였을 때, 분류를 고려하지 않은 기대치 값은 다음과 같다.

$$\sum_{i=1}^{I} W(A_i) \sum_{j=1}^{J} P(A_i = V_{ij})^2 \quad (4)$$

결과적으로, 본 논문에서 제안하는 수정된 Category Utility는 다음과 같다.

$$\frac{\sum_{k=1}^{K} P(C_k) \sum_{i=1}^{I} W(A_i) \sum_{j=1}^{J} [P(A_i = V_{ij} | C_k)]^2}{K} - \frac{\sum_{i=1}^{I} W(A_i) \sum_{j=1}^{J} P(A_i = V_{ij})^2}{K}$$

4.2 Weighted Etzioni 클러스터링

Etzioni의 클러스터링 방식은 전통적인 계층구조로서 클러스터의 질을 정량화하기 위해서 GQF(Global Quality Function)이라는 평가함수를 제안하고 있다[2]. Intersection-based 클러스터링 방식에서는 클러스터의 응집도를 정의하는 것이 핵심 문제라고 할 수 있다. 그리고 단일 클러스터의 스코어는 클러스터를 구성하는 문서의 수와 정규화된 클러스터 응집도의 곱으로 정의하고 $s(c)$로 표시한다. 아래의 수식은 클러스터 스코어를 계산하는 수식이다.

$$s(c) = |c| \times \frac{1 - e^{-\beta h(c)}}{1 + e^{-\beta h(c)}}$$

Weighted Feature의 중요도를 고려하기 위해서 본논문은 h(c)를 다음과 같이 정의한다.

$$h(c) = \sum_i W_i * T_i$$

$T_i = 1$ (If 클래스내에 attribute i의 Value가 같다.)

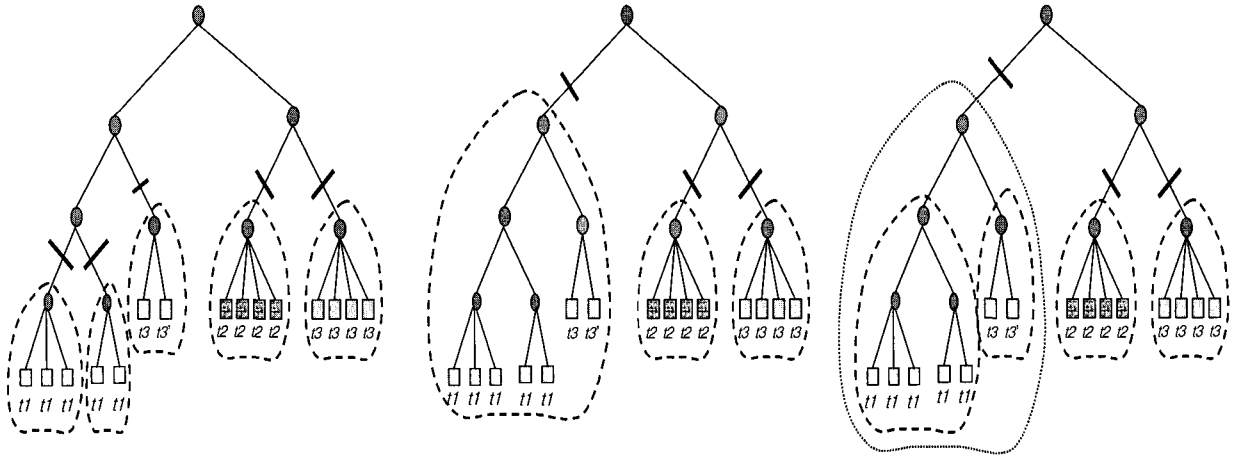
$T_i = 0$ (If 클래스내에 attribute i의 Value가 같지 않다.)

즉, h(c)는 클러스터 c의 응집도를 클러스터에 속하는 모든 데이터들에 공통적으로 나타나는 각 속성(feature)의 중요도 합으로 계산한다. 이때, β는 클러스터의 크기와 클러스터의 응집도 사이의 Trade-off를 정하기 위한 변수이다. β값은 경험적으로 0에서 1사이의 값을 가지는 것이 바람직하다.

5. 초기 클러스터의 평가 함수 제안

COBWEB이 생성한 분류 트리는 데이터들 간의 유사도를 표현하는 구조이며, 이 분류 트리로부터 클러스터를 생성하기 위해서는 트리를 임계값을 기준으로 나누어 주는 것이 필요하다. 최종 클러스터의 정확성은 COBWEB이 생성한 분류 트리의 초기 클러스터의 정확도에 좌우되며, 최종 클러스터의 속도 또한 초기 클러스터의 크기에 따라 좌우된다. 그러므로, 초기 클러스터를 생성하기 위한 정확한 임계값을 정하기 위한 방법이 요구 된다. 다음은 COBWEB이 생성한 분류 트리에서 초기 클러스터를 생성한 세 가지 방법을 도식화 한 것이다.

(a)의 정적 방식은 COBWEB이 생성한 분류 트리에서 초기 클러스터를 생성하는 방법의 일종으로 평가함수를 적용하지 않고 정적으로(Static) 최하위 클러스터를 초기 클러스터를 생성하는 방식이다. COBWEB이 생성한 문서 분류 트리의 터미널 노드 위에서 각 클러스터들을 절단함으로써 초기 클러스터를 생성한다. (b)의 동적 방식은 초기 클러스터 생성을 위해서 평가함수를 이용하며 분류 트리의 임의의 노드에서 초기 클러스터 생성 임계값을 만족하면 동적으로 초기 클러스터를 생성한다. 결과적으로 초기 클러스터의 수를 줄이게 되어, 클러스터간 병합 연산을 줄일 수 있고 클러스터링 속도면에서 향상을 가져올 수 있었다. (c)는 (b)의 동적 방식과 유사 하지만, 분리된 서브 클러스터 내에서 단말 노드들간의 상호 관계를 다시 한번 파악하여, 초기 클러스터를 결정한다. (b)의 방법은 속도면에서 장점인 반면, 생성된 클러스터 내부에 노이즈(응집도가 떨어지는 데이터)가 있을 확률이 높다. 본 논문에서는 최소의 노이즈, 즉 응집도를 높이고, 계산 효율을 위해서 (c)의 방식을 제안한다. 초기 클러스터링을 위한 (c)의 방식은 두단계로 이루어진다. 첫 단계는 다음 평가 함수를 이용하여 계층적인 트리 구조를 클러스터 시키기 위한 Threshhold값을 정한다. 둘째, 클러스터내의 단말 노드간의 관계를 정의하여 클러스터의 분리 여부를 결정한다. 다음은 각각에 대해 자세히 설명하도록한다.



(a) (b) (c)

첫째, 다음은 초기 클러스터를 생성하기 위해서 Threshhold값을 정하기 위한 함수를 정의한 것이다.

$$\frac{S}{D} \times SplitInfo < \alpha, \quad SplitInfo = log\left(\frac{SizeOfNode}{SizeOfParentNode}\right)$$

위의 수식에서 S는 현재 노드에 속하는 문서들의 유사도를 표현하는 것으로 클래스 k에서 속성 값들의 표준편차의 합으로서 대입된다. D는 현재 노드에 속하는 문서들이 같은 레벨에 있는 노드에 속하는 문서들과 얼마나 유사하지 않는지를 표현하는 것으로 현재 노드의 부모 노드에서 속성 값들의 표준편차의 합으로써 대입된다. α는 실험에 의해 결정되는 초기 클러스터 생성을 위한 임계값이다. 각 노드에서의 표준편차의 합이 크다는 것은 하위노드들의 유사도가 낮다는 것을 의미한다. 분자인 S항은 하위노드의 문서들이 유사할수록 작은 값을 가진다. 반면에 분모인 D항은 현재 노드인 클래스가 형제 노드들과 다른 성격을 가질수록 큰 값을 가진다. 그러므로 현재 노드에서 보았을 때 하위노드들의 유사도가 높고 형제 노드들과 다른 점이 많은 경우에 그리고 현재 노드에 속하는 문서의 개수가 작을수록 전체적인 평가함수의 값을 감소시킨다

둘째, 클러스터내의 단말 노드간의 관계를 정의하여 클러스터의 분리 여부를 결정하기 위하여 다음과 같은 절차를 수행한다.

[1 단계] 가중치 테이블에 속해있는 각 속성들을 중요도 순으로 정렬시킨다.

[2 단계] 생성된 초기 클러스터에서 2개 이상의 구성요소를 지니고 있는 클러스터에 대한 응집도에 대한 데이터 검사를 시작한다.

① 가중치 테이블에서 가장 높은 중요도를 나타내는 속성값에 대하여 분류화(Classify) 작업을 수행한다. (가중치 '0, 1'은 제외)

② 분류화 작업의 결과에 따라서 다음의 세 경우로 나뉘어 진다.

- Split, One일 경우는 다음으로 고려되는 속성에 대한 분류화 검사를 진행한다.
- 완전 Binary의 경우는 다음으로 고려되는 속성에 대한 분류화 검사를 진행하며 더 이상 고려되는 속성값이 없을 때까지 완전 Binary 형태를 유지한다면 이를 분리하여 새로운 클러스터로 생성한다.
- 편향된 Binary의 경우는 해당 속성에 대하여 클러스터를 분류화 하고 새로운 클러스터로 생성시킨 후 각 클러스터에 대하여 다음으로 고려되는 속성에 대한 분류화 검사를 진행한다.

③ 더 이상 고려되는 클러스터나 속성값이 없을 때까지 ①~②의 수행을 반복한다.

이러한 속성값에 대한 분류화 작업은 사용자가 부여한 속성값의 가중치에 중점을 두고 클러스터내의 응집도가 낮은 데이터를 판별하고 분리해 내는 역할을 한다. 이처럼 초기 클러스터가 생성되면, 최종 클러스터 병합을 위한 Etzioni 방식의 클러스터 병합 과정을 거쳐 최종 클러스터로 생성되게 된다.

6. 실 험

본 논문에서 제안한 방식의 우수성을 증명하기 위하여 다음과 같은 클러스터링 성능 비교 실험을 실시하였다. 먼저 본 논문은 Cobweb + Etzioni 통합 클러스처링 방식에 그 기반을 두고 있다[12]. 그렇기 때문에 비교실험 대상은 본 논문에서 제안하는 방식과 순수한 Cobweb + Etzioni 통합 클러스처링 방식간의 성능비교 실험을 실시하여 우수성을 증명하겠다. 실험을 위한 입력 자료는 가상의 인터넷 쇼핑몰을 구축하여 사용하였다. 입력 데이터를 얻기 위하여 가상 쇼핑몰 상에서 사용자가 관심을 보이는 제품 정보를 모

니터하고 이를 기반으로 제품정보에 대한 데이터를 추출하여 실험 데이터 집합을 생성한다.

사용자는 이 쇼핑몰에 로그인을 하여 자신이 관심 있어 하는 상품에 대하여 브라우징이나 구매 같은 행위를 하게 된다. 이러한 사용자 행위는 모니터 에이전트를 통하여 사용자 행위 DB에 저장이 되며 이를 기반으로 입력 데이터가 생성된다. 이렇게 생성된 실험 데이터는 총 24개 카테고리에 274개의 제품 정보를 담고 있다. 이 입력 데이터를 기반으로 총 6회 실험을 실시하였다.

실험의 목적은 본 논문에서는 제안하는 방식과 Cobweb + Etzioni 클러스터링 방식을 비교함으로써 제안하는 방식이 더욱 우수함을 증명하려 한다. 이를 위하여 각 클러스터링 결과에 대한 성능 평가를 위한 측도(Measure)를 정의하였다. 현재 클러스터링 결과에 대한 성능 평가와 관련해서 많은 연구가 진행되고 있지만 정확한 평가기준은 정해지지 않은 상태이므로 일반적으로 정보 검색 분야에서 많이 사용되고 있는 평가 측도들을 이용하여 각 클러스터링 방식의 성능을 비교하도록 하였다[9]. 실험에서 사용한 평가 측도들은 다음과 같다.

$$\text{① 평균 정확도} = \frac{\sum E_i(\sum C_j(|d|)/T)}{N}$$

평균 정확도는 각 실험에서의 정확도의 평균으로, 정확도 계산은 전체 문서 중에서 최종 클러스터에 맞게 분류된 문서들의 비율로서 계산한다. 평균 정확도의 값이 높은 클러스터링 방식이 성능 면에서 더욱 우수하다고 할 수 있다. 위의 수식에서 C_j는 최종 클러스터들을 의미하며, d는 맞게 분류된 문서, T는 전체 문서의 개수를 의미한다. 그리고 E_i는 각 실험을 의미하고, N은 전체 실험 횟수를 나타낸다. 이러한 정확도의 계산은 프로그램상으로 할 수 없었기 때문에 직접 클러스터링 결과를 검사하고 측정하는 방식을 취하였다.

$$\text{② 평균 클러스터 개수 오차} = \frac{\sum E_i(|C_O - C_I|)}{N}$$

클러스터 개수 오차는 실험의 입력 카테고리 개수와 출력 카테고리 개수와의 차를 의미하여, 평균 클러스터 개수 오차를 평가 측도로서 사용한다. 입력 카테고리 개수와 출력 카테고리 개수의 차이가 적을수록 의도한 클러스터링이 수행되었다는 의미이므로 평균 클러스터 개수 오차의 값이 작을수록 더욱 좋은 결과로 평가할 수 있다. 위의 수식에서 C_O는 출력 카테고리 개수를 의미하고, C_I는 입력 카테고리 개수를 의미한다. E_i는 각 실험을 표시하며, N는 전체 실험횟수를 나타낸다.

$$\text{③ 평균 실행 속도} = \frac{\sum E_i(|T_s - T_f|)}{N}$$

실행 속도는 각 클러스터링이 시작된 시각과 끝난 시각의 차를 계산하여 평가측도로 사용하였다. 본 논문에서는 기존의 Cobweb + Etzioni 방식에 추가적으로 가중치를 부여하고 노이즈 제거방식을 사용하였다. 그렇기 때문에 제안하는 방식의 수행 시간이 최소한 비슷하거나 작게 측정되어야지만 우수한 방식이라고 할 수 있겠다. 위의 수식에서 T_s는 클러스터링을 시작한 시각이고, T_f는 클러스터링이 완료된 시각이다. E_i는 각 실험을 의미하고, N은 실험 횟수를 의미한다.

다음의 표는 본 논문에서 제안하는 방식과 성능비교 실험의 대상인 Cobweb + Etzioni 방식의 실험 결과 데이터이다.

〈표 1〉 논문에서 제안한 방식의 실험 결과

실 험	정확도(%)	개수 오차	실행 속도(sec)
1	100	0	2.28
2	97	0	2.51
3	75	0	1.79
4	100	0	2.09
5	96	1	1.95
6	100	0	1.9

〈표 2〉 Cobweb + Etzioni 통합 방식의 실험 결과

실 험	정확도(%)	개수 오차	실행 속도(sec)
1	96	0	2.09
2	94	1	2.93
3	66	2	1.93
4	89	1	2.15
5	63	3	2.37
6	69	1	2.21

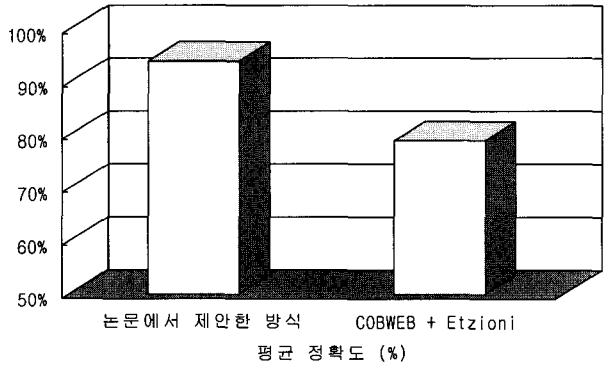
먼저, 다음 그래프는 평균 정확도 측면에서 비교한 그래프이다. 노이즈 제거 방식은 평균 정확도가 94.66%, Cobweb + Etzioni 방식은 79.5%로 큰 차이를 보이고 있다. 제안하는 방식이 더욱 우수한 성능을 나타나고 있다.

다음은 평균 클러스터 개수 오차에 대해서 알아보겠다. 제안하는 방식은 0.16의 수치를 나타내었고 Cobweb + Etzioni 방식은 1.33의 수치를 보였다. 역시 제안하는 방식이 더욱 우수한 성능을 나타내고 있다.

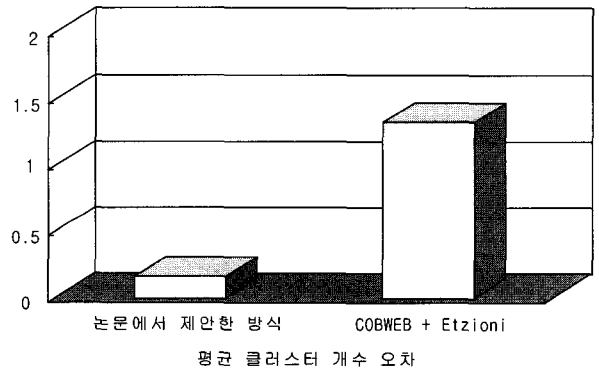
마지막으로 평균 실행기간 비교 실험의 결과이다. 시행 시간은 두 방식 모두 별 차이가 나지 않았다. 제안하는 방식은 2.08(sec), Cobweb + Etzioni 방식은 2.28(sec)의 시간이 평균으로 산출되었다.

실험결과 본 논문에서 제안하는 방식이 평균 정확도와, 평균 개수 오차 측면에서 더욱 우수한 성능을 나타내었다. 이러한 이유는 응집도가 높은 클러스터를 초기 클러스터로 분류해냈으며, 가중치에 근거한 클러스터를 수행함으로써

최종 클러스터의 정확성을 높였기 때문이라고 분석된다. 또한 평균 실행 시간에서는 두 방식의 실행 시간이 거의 차이가 나지 않았다. 이것은 제안하는 방식에 포함된 가중치 기반 클러스터링과 최종 클러스터를 생성하는 작업들이 모든 수행시간에 별 영향을 미치지 않고 효율적으로 수행되었다는 것을 증명하는 것이다.



(그림 3) 평균 정확도 비교 결과



(그림 4) 평균 클러스터 개수 오차 비교

7. 결 론

통합 클러스터링 방식은 먼저, COBWEB 알고리즘을 이용하여 각 데이터간의 상호 연관성을 정의하고, 상호 연관성을 기준으로 초기 클러스터를 생성한다. 초기 클러스터들은 Etzioni 알고리즘을 이용하여 최종 클러스터링을 생성하는 방식이다. 본 논문에서는 대용량의 데이터를 효율적으로 클러스터링하기 위해 사용되는 점진적 클러스터링 방법의 단점인 입력 의존적인 문제를 효율적으로 해결하기 위하여 초기 클러스터의 평가 함수를 제안한다. 제안한 초기 클러스터 생성 평가함수를 이용하여 클러스터내의 응집도를 높여, 최종적으로 보다 나은 클러스터가 되도록 하였다. 그리고, 클러스터할 데이터내의 속성의 중요도를 고려하기 위하여, 통합 클러스터내의 COBWEB, Etzioni의 평가 함수를 수정하였다. 본 논문에서 제안한 가중치를 고려한 통합 클러스터링 방식은 웹에이전트나 쇼핑 에이전트에서 사용자의 프로파일을 생성하는데 유용하게 사용될 수 있다.

앞으로, COBWEB의 분류 트리에서 클러스터내의 데이터 응집도가 높으면서 충분한 크기의 초기 클러스터가 되는 임계값을 구하기위한 연구를 계속 수행해 나갈 것이다. 그리고 가중치 기반의 클러스링방식은 현재 사용자가 클러스터할 데이터의 속성의 중요도를 직접 넣고 있으나 웹 에이전트같은 응용 시스템에서 자동적으로 조절할 수 있도록하려고 한다. 즉, 사용자의 관심을 파악하기 위하여 클러스터링이 사용되나, 클러스터링의 결과가 다시 피드백으로, 클러스터할 데이터의 속성의 중요도에 영향을 주게 되는 것이다. 이때, 사용자의 지속적인 관심을 중심으로 데이터를 클러스터를 함으로, 클러스터와 사용자 관심 파악의 효율을 높이고자 한다.

《감사의 글》
본 논문을 쓰는데 도움을 준 양찬범, 이성열 후배에게 감사드립니다.

참 고 문 헌

[1] Mark Devaney, Ashwin Ram, "*Efficient Feature Selection in Conceptual Clustring*," Machine Learning : Proceeding of the Fourteenth International Conference, Nashville, 1997.
[2] Oren Zamir, Oren Etzioni, Omid Madani and Richard M. Karp, "*Fast and Intuitive Clustering of Web Documents*," KDD '97, 1997.
[3] Doug Fisher, "*Interative Optimization and Simplification of Hierarchical Clusterings*," AI Access foundation and Morgan Kaufmann Publishers, 1996.
[4] Gennari, J. H., Langley, P. & Fisher, D. H., "*Models of incremental concept formation*," Artificial Intelligence, 40, pp.11-61, 1989.
[5] Gluck, M. & Corter, J., "*Information, uncertainty and the utility of categories*," Proceedings of the Seventh Annual Conference of the Cognitive Science Society, pp.283-287, Irvine, CA : Lawrence Erlbaum, 1985.
[6] Hartigan, J.A., "*Clustering Algorithms*," Wiley, New York, 1975.
[7] T. M. Mitchell, "*Machine Learning*," McGraw Hill, 1997.
[8] Kathleen Mckusick, Kevin Thompson, "*COBWEB/3 : A Portable Implementation*," NASA Ames Research Center, Technical Report FIA-90-6-18-2, 1990.
[9] Robert R. Korfhage, "*Information Storage and Retrieval*," Wiley Computer Publishing, 1997.
[10] Richard C. Dubes and Anil K. Jain., "*Algorithms for Clustering Data*," Prentice Hall, 1988.
[11] Wettschereck, D., Aha, D. W. & Mohri, T. "*A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms*," Artificial Intelligence Review, 11, pp.273-314, 1997.
[12] 양찬범, "*웹 에이전트를 위한 통합방식 문서 클러스터링*", 숭실대학교 석사학위논문, 1999.

백 혜 정

e-mail : hjbaek@multi.soongsil.ac.kr

1995년 숭실대학교 컴퓨터학과(학사)

1998년 숭실대학교 대학원 컴퓨터학과(공학석사)

1999년~현재 숭실대학교 대학원 컴퓨터학과 박사과정

관심분야 : 인공지능, 에이전트, 전문가 시스템

박 영 택

e-mail : park@computing.soongsil.ac.kr

1978년 서울대학교 전자공학과(학사)

1980년 KAIST 전산학(석사)

1992년 Univ. of Illinois at Urbana-Champaign(박사)

1981년~현재 숭실대학교 컴퓨터학과 교수

관심분야 : 인공지능, 에이전트, 전문가 시스템