논문 2012-49CI-3-1

유전자 알고리즘을 통한 XML 군집화 방법

(XML Clustering Technique by Genetic Algorithm)

김 우 생*

(Woosaeng Kim)

요 약

최근 들어 인터넷에서 많이 사용되는XML 문서들을 효율적으로 접근, 질의, 관리하는 방법들이 연구되고 있다. 본 논문은 XML 문서들을 효율적으로 군집화 하는 새로운 기법을 제안한다. XML 문서의 원소는 대응하는 트리의 노드에 대응하며, 문서에서 내포 관계는 트리의 부모와 자식 노드 간의 관계에 대응한다. 따라서 유사한 XML 문서들은 대응하는 트리들에서 노드의 이름과 레벨 등이 유사하다. 이러한 성질을 유전 알고리즘의 평가 함수로 만들어 군집화를 시도하였다. 실험 결과를 통하여 제안하는 기법이 기존 방법들보다 좋은 결과를 얻을 수 있음을 보였다.

Abstract

Recently, researches are studied in developing efficient techniques for accessing, querying, and managing XML documents which are frequently used in the Internet. In this paper, we propose a new method to cluster XML documents efficiently. An element of a XML document corresponds to a node of the corresponding tree and an inclusion relationship of the document corresponds to a relationship between parent and child node of the tree. Therefore, similar XML documents are similar to the node's name and level of the corresponding trees. We make evaluation function with this characteristic to cluster XML documents by genetic algorithm. The experiment shows that our proposed method has better performance than other existing methods.

Keywords : XML 군집화(XML Clustering), 유전 알고리즘(Genetic algorithm)

Ⅰ. 서 론

인터넷의 성장은 전 세계에 존재하는 모든 데이터와 정보의 접근을 쉽게 만들면서 많은 데이터들이 다양한 형태의 정보로 생성되는데 이바지하고 있다. 인터넷이 점점 성장하고 발전할수록, 더 많은 정보들은 XML과 같이 구조적으로 풍부한 문서 형태로 존재하게 된다. 웹에서 문서가 많아질수록, 이와 같이 구조적으로 풍부한 문서들을 자동적으로 검색하고 관리하는 응용들이 요구되고 있다.

XML 문서들에 대한 군집화는 유사한 문서들의 그룹을 만들어 특정한 카타고리 안에서 검색과 처리를 용이하게 하기 위함이다. XML 문서에 대한 적절한 군집화는 체계적인 문서 관리와 문서 저장을 위해서도 효율적이다. 또한 군집화 된 데이터들은 데이터들 간에 일종의 경향 또는 규칙성을 보이고 심지어 주목할 가치가 있는 관련 지식을 보여 주기까지 한다.

본 논문은 유전 알고리즘을 사용하여 XML 문서들을 효율적으로 군집화 하는 방법을 제안한다. XML문서의 원소들은 내포 구조로 구성되기 때문에, XML은 정렬된 라벨 트리로 모델링 될 수 있다[1]. 문서의 원소는 트리의 노드에 대응하며, 문서에서 내포 관계는 트리의 부모와 자식 노드 간의 관계에 대응한다. 본 논문은 같은 DTD에서 생성된 XML 문서들은 유사한 문서들로 간주한다. 따라서 유사한 XML 문서들은 대응하는 트리들의 노드 이름과 레벨 등이 유사하며, 이에 기반한 평가

* 정회원, 광운대학교 컴퓨터소프트웨어
(Dept. of Computer Software, Kwangwoon Univ.)
※ 본 논문은 2012년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.
접수일자: 2012년3월15일, 수정완료일: 2012년5월4일

함수를 사용하는 유전 알고리즘을 설계하였다. 실험을 통해 이 방법이 기존의 방법들보다 군집화가 잘 이루어짐을 보인다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 XML 군집화 관련 연구에 대해 기술하고, Ⅲ장은 XML 문서 군집화를 위한 유전 알고리즘을 제안한다. Ⅳ장은 실제 데이터를 통한 실험을 통하여 제안한 방법이 효율적인지를 조사한다. 마지막으로 Ⅴ장은 결론을 낸다.

Ⅱ. 관련 연구

1. XML 군집화

다양한 XML의 증가로 인해 XML 데이터를 조직하고 군집화 하는 필요성이 증대되고 있다. 최근에 XML 문서들의 구조뿐 아니라 내용을 고려하는 군집화 기법들이 연구되고 있다. 문서간의 구조에 대한 공통 구조의 존재 여부를 0, 1로 표현하는 비트를 이용하여 비트맵 인덱스에 기반한 군집화 기법이 제안되었다[2~3]. BitCube는 3 요소 즉, 문서, 경로, 단어의 3 차원 비트맵 인덱스로 표현된다. BitCube 인덱스들은 문서들을 분할해서 군집화하기 위해 bit-wise 거리 척도를 활용한다. 그러나 이 방법은 비트맵 인덱스를 생성하기 위해서 수작업을 필요로 한다. XML 데이터를 위한 특징들로 문서로부터 추출한 내용 정보와 태그 경로들로부터 유도되는 구조 정보들을 사용하는 방법이 제안되었다[4]. 이 방법은 XML 문서 트리들을 트랜잭션 데이터 즉, 속성들을 가진 객체들로 사상하는 것을 허용하는 XML 표현 모델의 정의 안에서 트리 투플이라는 개념을 소개하며, 군집화 기법이 XML 트랜잭션의 영역 안에서 개발되고 적용되었다. 반면에 XML 문서의 구조를 이산 함수로 변환하는 방법이 연구되었다[5]. 이산 함수는 FFT에 의해서 주파수 영역으로 변환된다. FFT의 결과는 x와y의 값들을 포함하는 복소수 쌍들이며 n 차원 벡터들로 간주되어 유클리디안 거리 척도를 사용하여 비교된다. 이 방법은 오직 문서들의 구조만을 고려한다. 다양한 구조를 가지는 XML 문서의 경로 구조를 중심으로 빈발 고조에 대한 유사성 기반의 점진적 클러스터링 기법을 제안되었다[6]. XML 문서를 구성하는 원소의 순서와 발생 빈도를 동시에 고려할 수 있는 순차 패턴을 이용하여 일정한 지지도를 만족하는 빈발 구조 패턴을 추출하여 유사 구조 문서를 그룹화 하여 주요 항목 기반의 클러스터를 생성하고, 클러스터 할당 이익에 대한 연산을 통해 점진적 클러스터링을 수행하였다. XML 문서를 대응하는 트리 구조의 원소들의 이름과 레벨의 n차원 특징 벡터 $x=[x_1x_2...x_n]^T$로 표현한 후, 이러한 특징 벡터 공간 상의 XML 문서들에 주성분 분석을 적용한 k 평균 알고리즘을 사용하는 군집화 기법이 제안 되었다[7].

2. 유전알고리즘

유전 알고리즘은 크게, 문제의 해를 표현하는 염색체(chromosome), 염색체 다수로 구성된 해집단(population), 해의 적합도를 평가하는 평가 함수(evaluation function)로 구성되며, 이 해들을 선택(selection), 교배(crossover), 변이(mutation)의 과정을 거쳐 개선시켜 나감으로써 점차 최적에 가까운 해를 얻게 된다[8]. 기존 알고리즘은 하나의 초기 해를 생성한 후 적절한 연산을 사용해 이것을 조금씩 개선해 나가나, 유전 알고리즘은 초기 해를 여러 개 생성해 해집단을 구성하고 이 해집단을 개선해 나가는 점이 다르다.

유전 알고리즘의 전체적인 흐름은 그림 1과 같고 각 단계는 다음과 같다. 어떤 문제를 푸는데 유전 알고리즘을 적용하기 위해서는 먼저 문제의 해를 적절히 부호화해서 표현해야 한다. 부호화 표현 방법에는 실수, 이진수 등 여러 가지 표현 방법이 존재한다. 이러한 일정 개수의 해들로 초기 해집단을 구성하게 되고, 이 해집단에 선택, 교배, 변이와 같은 유전 연산자들을 적용해 점차 해를 개선시켜 나간다. 초기 해집단에서 선택 연산을 통해 두 개의 부모 해가 선택되면 두 부모 해를

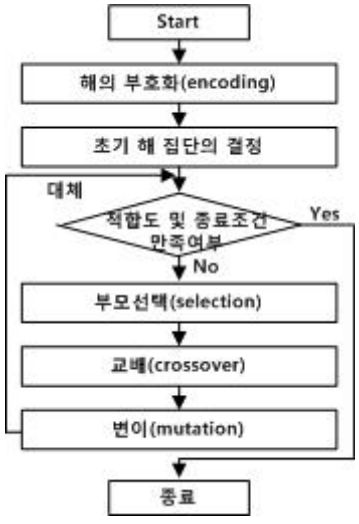


그림 1. 유전 알고리즘의 일반적인 구조
Fig. 1. The general structure of genetic algorithm.

추출하여 자식 해를 얻는 교배 연산을 적용하게 된다. 이렇게 생성된 자식 해는 해의 다양성을 위해 부모 해에 없는 유전자를 생성하는 변이 연산을 거쳐 최종적인 자식 해가 만들어 진다. 얻어 진 자식 해는 보통 해집단에서 가장 좋지 않은 해와 교체된다. 유전 알고리즘은 지금까지의 과정을 종료 조건을 만족될 때까지 반복하게 된다. 종료 조건은 다양한 방식이 있지만 대개의 경우 해의 적합도를 평가하여 해의 품질이 일정 정도에 이르게 될 때 종료하거나 특정 세대 수만큼 진화 연산을 적용한 후 종료하는 방식을 취한다.

Ⅲ. XML 문서 군집화 위한 유전 알고리즘

DTD에서 생성된 유사한 문서들을 군집화하기 위한 유전 알고리즘의 구조는 그림 2와 같이 구성된다. 유전 알고리즘의 각 단계에 대한 설명은 이 장의 소절에서 각각 설명한다.

```
n개의 초기 염색체 생성
생성된 염색체들로 해집단 구성
해집단의 각 해를 평가 함수로 평가
repeat {
    해집단에서 두 부모 해 p1, p2선택
    부모 해 p1, p2를 교배해 자식 해 생성
    자식 해를 변이시킴
    자식 해 평가 함수로 평가해 해집단
    내의 한 해와 대체
} until (종료 조건);
return 해집단에서 가장 좋은 해
```

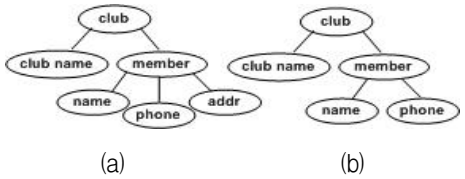
그림 2. 유전 알고리즘의 구조
Fig. 2. The structure of genetic algorithm.

1. 해의 표현

그림 3의 Club의 DTD에 의해 생성되는 일부 문서들을 대응하는 트리로 표현하면 그림 4의 (a), (b)로 표현된다. 예를 들어, 그림 4(a)는 트리의 첫 레벨에 있는 노드 club, 두 번째 레벨에 있는 노드들인 clubname, member 그리고 세 번째 레벨에 있는 노드들인 name, phone, addr로 표현된다. XML 문서를 대응하는 트리 구조의 노드들의 이름과 레벨로 표현할 때, 같은 DTD에 의해 생성되는 유사한 문서들은 같은 레벨의 같은 노드 이름들을 많이 공유함을 알 수 있다. 예를 들어,

```
Club
 <!ELEMENT club (clubname, member+)>
 <!ELEMENT clubname (#PCDATA)>
 <!ELEMENT member (name, phone, addr?)>
 <!ELEMENT name (#PCDATA)>
 <!ELEMENT phone (#PCDATA)>
 <!ELEMENT addr (#PCDATA)>
```

그림 3. Club DTD
Fig. 3. Club DTD.



(a) (b)

그림 4. Club DTD에 대응하는 트리들
Fig. 4. Trees correspond to Club DTD.

Club DTD에 의해 생성되는 그림 4(a), (b)는 첫 번째 레벨의 club은 루트로 같고, 두 번째 레벨의 clubname과 member, 그리고 세 번째 레벨의 name과 phone이 같다.

XML 문서를 대응하는 트리 구조의 정보로 표현하기 위해, 본 논문에서는 모든 XML 문서에 포함된 원소들을 알파벳 순서로 정렬한 후 각 원소에 대응되는 루트 값과 레벨 값으로 표현한다. 여기서 원소의 루트 값과 레벨 값은 각각 원소가 소속된 문서의 루트 노드의 정렬된 위치와 원소에 대응하는 노드의 레벨 위치이다 만약 한 XML 문서에 같은 원소가 여러 레벨에 걸쳐 존재하면, 레벨 값은 루트에 가장 가까운 레벨의 위치이다.*

예를 들어, 그림 4(a), (b)에 대응하는 XML 문서는 표1의 X_1, X_2로 표현된다. X_1의 원소 addr의 루트 값은 루트 노드인 club의 정렬된 위치인 2이고, 레벨 값은 3이기 때문에 (2,3)으로 표현되고, 원소 club의 루트 값은 2 레벨 값은 1이기 때문에 (2,1)로 표현된다. X_1의 나머지 4개의 원소들도 이와 같은 방법으로 표현된다. X_2의 경우는 원소 addr을 포함하고 있지 않기 때문에 원소 addr은 (2, '-')로 표현된다. 유사한 문서들의 경우 DTD의 특별한 기호인 *, ?가 붙지 않은 원소의 루트 값과 레벨 값은 항상 같다. 본 논문에서는 원소의 루트 값과 레벨 값이 모두 같을 때의 원소를 동일 원소, 그

* 만약 한 XML 문서에 같은 원소가 여러 레벨에 걸쳐 존재하면, 레벨 값은 루트에 가장 가까운 레벨의 위치이다.

표 1. 트리에 대응하는 XML 문서 표현
Table 1. XML document representation corresponding to the tree.

	addr	club	cname	mem	name	pho
X_1	(2,3)	(2,1)	(2,2)	(2,2)	(2,3)	(2,3)
X_2	(2,-)	(2,1)	(2,2)	(2,2)	(2,3)	(2,3)

외의 경우, 즉 원소의 루트 값이 다르거나 레벨 값이 다를 때의 원소를 비동일 원소라 칭한다. 표 1에서 원소 addr는 비동일 원소이고 나머지 원소들은 모두 동일 원소이다.

유전 알고리즘을 사용하기 위해서는 먼저 군집화 하고자 하는 XML 문서들을 염색체로 표현해야 한다. 문서 $X_1...X_m$을 n개의 그룹으로 군집화 할 때 염색체는 m개의 정수들로 표현되며, i번 째 정수 C_i(단 $1 \le C_i \le n$)는 문서 Xi가 할당된 그룹 번호가 된다. 예를 들어, 염색체 (1 1 2)는 두 개의 군집 $\{X_1, X_2\}$와 $\{X_3\}$를 의미한다. 무작위로 선출 된 이러한 다수의 염색체들로 초기 해집단을 구성한다.

2. 평가 함수

각 해의 적합도를 구하기 위해 그 품질을 평가해야 하는데, 품질의 평가는 얼마나 많은 문서들이 동일 원소를 갖고 있는가를 고려한다. Actor DTD가 그림 5와 같을 때, 표 2는 Actor와 Club의 DTD에 의해 생성된 일부 XML 문서들의 원소들을 알파벳 순서로 정렬한 후 각 원소에 대응되는 루트 값과 레벨 값으로 표현한 것이다.

표 2에서 유사한 문서들로만 군집화가 될 때, 즉 $\{A_1, A_2, A_3,\}$와 $\{C_1, C_2\}$로 군집이 이루어 질 때 첫 번째 군집에서 동일 원소의 수는 4이고 두 번째 군집에서 동일 원소의 수는 5로 전체 동일 원소의 수는 9이다. 그러나 다른 조합으로 군집화를 하면 적어도 한 군집은

```
Actor
 <!ELEMENT actor (name, addr?, movie)>
 <!ELEMENT name (#PCDATA)>
 <!ELEMENT addr (#PCDATA)>
 <!ELEMENT movie (title, year?)>
 <!ELEMENT title (#PCDATA)>
 <!ELEMENT year (#PCDATA)>
```

그림 5. Actor DTD
Fig. 5. Actor DTD.

표 2. 트리에 대응하는 XML 문서 표현
Table 2. XML document representation corresponding to the tree.

	act	addr	clu	cna	mem	mov	name	pho	tit	yr
A_1	(1,1)	(1,2)	(1,-)	(1,-)	(1,-)	(1,2)	(1,2)	(1,-)	(1,3)	(1,3)
A_2	(1,1)	(1,-)	(1,-)	(1,-)	(1,-)	(1,2)	(1,2)	(1,-)	(1,3)	(1,3)
A_3	(1,1)	(1,-)	(1,-)	(1,-)	(1,-)	(1,2)	(1,2)	(1,-)	(1,3)	(1,-)
C_1	(3,-)	(3,3)	(3,1)	(3,2)	(3,2)	(3,-)	(3,3)	(3,3)	(3,-)	(3,-)
C_2	(3,-)	(3,-)	(3,1)	(3,2)	(3,2)	(3,-)	(3,3)	(3,3)	(3,-)	(3,-)

서로 유사하지 않은 문서들과 군집을 이루어야 하므로, 그 군집에 존재하던 동일 원소가 비동일 원소로 되어 전체 동일 원소의 수는 줄어든다. 예를 들어, $\{A_1, A_2\}$와 $\{A_3, C_1, C_2\}$로 군집화를 하면 $\{C_1, C_2\}$에 존재하는 모든 동일 원소가 비동일 원소로 되면서 전체 동일 원소가 4로 줄어든다. 즉, 유사한 문서들로만 군집이 이루어질 때 가장 많은 동일 원소가 존재한다. 이를 바탕으로 해의 평가 함수는 식 (1)로 표현된다. 여기서 동일원소_문서 수는 동일 원소를 포함하는 문서들의 수이며, m은 동일 원소 수, n은 군집의 숫자이다.

$$\sum_{\text{군집}=1}^{n} \sum_{\text{동일원소}=1}^{m} \text{동일 원소\_문서수} \qquad (1)$$

표 2의 경우 예를 들어, 군집이 $\{A_1, A_2, A_3\}$, $\{C_1, C_2\}$로 형성 될 때 첫 번째 군집의 동일원소_문서 수는 12이며 두 번째 군집의 동일 원소_문서수가 10이기 때문에 평가 함수는 22이다. 표 2에서 다른 어떤 조합의 군집화도 평가 함수는 22보다 작은 것을 알 수 있으며, 일반적으로 유사한 문서들로만 군집이 이루어질 때의 평가 함수가 가장 높다.

3. 선택 연산

평가 함수로 평가된 해들 중에서 교배에 사용할 두 개의 부모 해를 선택하는 방법으로는 해의 적합도에 비례하여 선택하는 룰렛휠(roulette-wheel) 방법을 사용한다. 하지만 주어진 문제에서의 해의 품질을 그대로 적합도로 사용하면 대부분의 경우 해집단에서 가장 우수한 해와 가장 열등한 해의 품질이 차이가 많이 나서 열등한 해들은 거의 선택될 기회를 잃게 된다. 이러한 문제점을 해결하기 위해서 각 해의 품질을 평가한 다음 가장 우수한 해의 적합도가 가장 열등한 해의 적합도보다 k 배가 되도록 조절한다. 해집단 내의 i번 째 해의 적합도는 식 (2)와 같이 계산된다. 여기서 C_w, C_b, C_i는 각각 해집단 내에서 가장 열등한 해, 가장 우수한 해, i

번 째 해의 평가 함수 값이다.

$$f_i = (C_w - C_i) + (C_w - C_b)/(k-1),\ k > 1 \qquad (2)$$

위의 방법을 사용하면 가장 좋은 해의 적합도가 가장 나쁜 해의 적합도보다 k배가 되도록 조절할 수 있다. 본 논문의 실험에서는 일반적으로 가장 흔히 쓰는 k값인 3을 사용한다.

4. 교배 연산

두 개의 부모 해가 선택이 되면 두 부모 해를 섞어 새로운 자식 해를 얻는 교배 연산을 수행하게 되는데, 교배 연산 방법은 균등 교배(uniform crossover) 방법을 사용한다. 균등 교배는 먼저 임계 확률 p를 설정한 후 각각의 유전자 위치에서 난수를 발생해 이 값이 p이상이면 부모 해 p_1의 같은 위치로부터 유전자를 복사하고, 그렇지 않으면 부모 해 p_2의 같은 위치로부터 복사한다.

그림 6에서 두 개의 부모 해 p_1, p_2가 선택되면 먼저 자식 해가 가지게 되는 첫 유전자를 p_1, p_2중에서 선택하기 위해 0에서 1 사이의 난수를 발생시킨다. 여기서는 난수로 0.24가 생성이 되었고, 이것은 임계 확률 p보다 작기 때문에 자식의 첫 유전자는 p_2의 처음 위치의 유전자를 가져오게 된다. 자식 해의 두 번째 유전자의 경우에는 발생한 난수가 0.81로 임계 확률 p보다 크기 때문에 p_1의 두 번째 위치에서 유전자를 가져오게 된다. 이 과정을 유전자의 수만큼 반복하면 p_1, p_2가 교배된 자식 해를 얻을 수 있다.

부모해 P_1: (2,3) (1,5) (3,2) (4,1) (4,4)
부모해 P_2: (5,4) (2,5) (5,1) (3,3) (2,1)
난수: 0.24 0.81 0.33 0.19 0.66 (=0.5)
자식해: (5,4) (1,5) (5,1) (3,3) (4,4)

그림 6. 균등 교배 적용의 예
Fig. 6. An example of uniform crossover.

5. 변이 연산

교배는 해집단 내에서의 해의 진화에 한계에 있다. 다시 말해 주어진 환경에 어느 한계까지는 진화하여 적응할 수 있지만, 부모 해에 없는 유전자를 도입하기는 힘들다. 예를 들어, 11110과 11100의 두 해가 아무리 교배를 한다 하더라도 11111이라는 해는 생길 수 없다. 변이는 해 내의 유전자를 직접 바꾸어 이러한 문제점을 보완한다. 이를 위해 교배에 의해 생성된 자식 해에서 변이시킬 유전자를 선택하기 위해 각각의 유전자의 위치에서 [0,1] 범위의 난수를 만들어 미리 정한 임계 값보다 작은 경우의 유전자만 변이시킨다. 이와 같이 변이 연산은 보다 다양한 해의 생성에 기여해 유전자 알고리즘이 더 넓은 공간을 탐색할 수 있도록 하기 때문에 지역 최적점에서 벗어나도록 하는 역할을 한다. 변이의 확률을 높이면 보다 다양한 해를 생성해 낼 수 있어 유전 알고리즘의 역동성은 늘어나지만 해집단의 수렴 시간이 오래 걸리는 경향이 있다. 본 논문에서는 변이의 확률로 0.01을 사용한다.

Ⅳ. 실험 결과

본 연구는 위스콘신 대학의 XML 데이터 뱅크에서 제공하는 데이터들을 사용하여 제안하는 방법의 효율성을 실험하였다. 이 데이터 뱅크는 movies, department, bibliography, company profiles, stock quotes, club, personal information 등의 DTD 들을 제공하며 일부 DTD 들은 name, phone, address 등의 원소 이름들을 공유한다[9]. 군집화성공률을 조사하기 위해 2개의 군집에서부터 7개의 군집까지 총 6가지의 서로 다른 크기의 집단을 실험하였다. 예를 들어, 3개 군집은 위스콘신 대학의 DTD 중에서 같은 원소 이름을 많이 공유하는3개의 DTD를 선택하여, 각각의 DTD에 대해 생성된 10개씩의 XML 문서들로 군집화를 시도하였다.

그림 7은 해집단(population)의 사이즈를 다르게 했을 때의 군집화 성공률을 보여 주는데, 군집화 성공률은 유전 알고리즘을 10번씩 수행한 평균값으로 구했다. 유전 알고리즘은 각 세대마다 얼마나 많은 해를 사용하는가를 결정하는 해집단 사이즈에 많은 영향을 받는다. 그림 7에서와 같이 일반적으로 해집단의 사이즈가 작을 때 보다는 해집단의 사이즈가 클 때 성공률이 높아진다. 이것은 유전 알고리즘은 여러 개의 해들을 하나의 해집단으로 시작하여 확률론적인 방법으로 탐색을 하므로, 해집단의 사이즈가 클 때 탐색 시간은 더 걸리나 더 전역적인 최적 해를 찾을 가능성이 높아지기 때문이다. 군집의 크기가 작을 때에는 해집단의 사이즈와 상관없이 최적의 값을 보이나, 군집의 크기가 커질수록 해집단의 사이즈가 커져도 최적의 값을 갖지 못함을 알 수

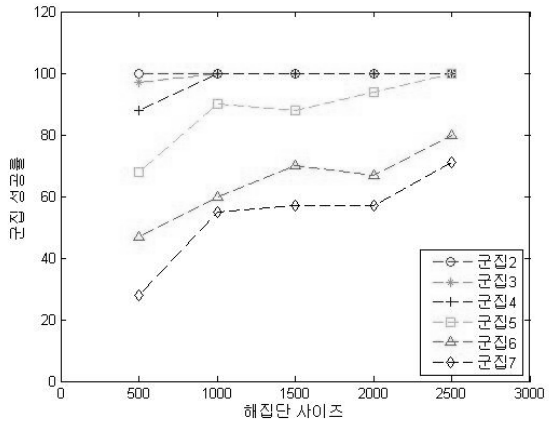


그림 7. 해집단 사이즈와 군집 수에 따른 군집 성공률
Fig. 7. Clustering success ratio corresponding to population size and cluster number.

표 3. 군집수에 따른 군집 성공률
Table 3. Clustering success ratio corresponding to the cluster number.

군집수	k 평균	주성분 k평균	유전자 (10번 평균)	유전자 (10 번 최적)
2	60	100	100	100
3	52	92	100	100
4	48	87	100	100
5	45	79	100	100
6	45	74	80	100
7	41	67	71	100

있다. 이것은 군집의 크기가 작을 때는 해의 크기와 동시에 해집단의 사이즈도 작아 전역적인 최적해를 찾을 수 있으나, 군집의 크기가 커질수록 해의 크기와 동시에 해집단의 사이즈도 커지므로 전역적인 최적해를 찾기 힘들어 지기 때문이다.

표 3은해집단의 사이즈를 2500으로 했을 때 제안 하는 방법과 기존의 방법과의 성능을 비교해 보여 준다[7]. 표에서 보듯이 제안하는 방법이 기존의 k 평균 알고리즘이나 주성분 분석을 통해 적절한 시드점을 찾는 k 평균 알고리즘 방법보다 성능이 좋음을 알 수 있다. 특히 유전 알고리즘을 10번 수행하여 가장 높은 평가 함수의 값으로 군집화를 시도할 때는 모두 전역적인 최적해를 구할 수 있었다. 따라서 군집화를 오프라인으로 수행할 수 있을 때는 기존 방법들 보다 유전자 알고리즘이 훨씬 효율적인 것을 알 수 있다.

V. 실험 결과

본 논문은 유전 알고리즘을 사용하여XML 문서들을 효율적으로 군집화 하는 방법을 제안하였다. 같은 DTD에서 생성된 유사한 XML 문서들은 대응하는 트리 구조도 유사하기 때문에, XML 문서를 대응하는 트리 구조의 노드들의 이름과 레벨 등으로 표현할 때 같은 DTD에 의해 생성되는 유사한 문서들은 같은 레벨의 같은 노드 이름들을 많이 공유한다. XML 문서를 대응하는 트리 구조의 정보로 표현하기 위해, 모든 XML 문서에 포함된 원소들을 알파벳 순서로 정렬한 후 각 원소에 대응되는 루트 값과 레벨 값으로 표현 하였다. 주어진 XML 문서들의 군집화가 제대로 이루어 질 때 가장 많은 동일 원소들을 가지며, 이를 평가 함수에 적용한 유전 알고리즘을 제안하였다. 실험 결과를 통하여 제안하는 기법이 기존의 k 평균이나 주성분 분석을 통한 k 평균 방법보다 좋은 결과를 얻을 수 있음을 보였다. 특히 유전자 알고리즘은 여러 번의 수행을 통하면 최적의 해를 얻을 가능성이 높아지기 때문에 오프라인에서 뛰어난 성능을 보인다.

참 고 문 헌

[1] R. Behrens, “A Grammar based model for XML schema integration,”Proc. of the 17th British National Conf. on Databases, pp.172-190, 2000.
[2] J. Yoon, V. Raghavan, V. Chakilam, “BitCube: clustering and statistical analysis for XML documents,” Proc. of the 13th Int. Conf. on Scientific and Statistical Database Management, Fairfax, Virginia, 2001.
[3] J. Yoon, V. Raghavan, V. Chakilam, L. Kerschberg, “BitCube: a 3-D bitmap indexing for XML documents,”Journal of Intelligent Information Systems, Vol. 17, pp.241-254, 2001.
[4] A. Tagarelli, A. Greco, “Toward semantic XML clustering,” 6th SIAM International Conference on Data Mining(SDM ’06), pp. 188-199. Bethesda, Maryland, USA, 2006.
[5] H. Lee, “An Unsupervised clustering technique of XML documents based on function transform and FFT,” Journal of Korea Information Processing Society, 2007.
[6] 황정희, 류근호 “유사 구조 기반 XML 문서의 점진적 클러스터링,” 정보과학회 논문지- 데이터베이스 제 31권 제 6호, 2004. 12.

[7] 김우생, “주성분 분석의 k 평균 알고리즘을 통한 XML 문서 군집화 기법,” 정보처리학회 논문지, 2011.10.
[8] 윤병로, “쉽게 배우는 유전 알고리즘,” 한빛미디어, 2008.4.
[9] Niagara Query Engine, http://www.cs.wisc.edu/niagara/data.html

저 자 소 개

김 우 생(정회원)-교신저자
대한전자공학회논문지,
제46권 CI편 제2호 참조