Journal of the Society of Korea Industrial and Systems Engineering
Vol, 26, No. 1, pp.17－21, March 2003.

범주형 값들이 순서를 가지고 있는 데이터들의 클러스터링 기법

오승준 · 김재련

한양대학교 산업공학과

Clustering Algorithm for Sequences of Categorical Values

Seung-Joon Oh · Jae-Yearn Kim

Dept. of Industrial Engineering, Hanyang University

We study clustering algorithm for sequences of categorical values. Clustering is a data mining problem that has received significant attention by the database community. Traditional clustering algorithms deal with numerical or categorical data points. However, there exist many important databases that store categorical data sequences. In this paper, we introduce new similarity measure and develop a hierarchical clustering algorithm. An experimental section shows performance of the proposed approach.

Keywords : clustering, sequences, categorical values

1. 서 론

클러스터링(Clustering)이란 물리적 혹은 추상적 객체들을 서로 비슷한 객체들의 집합으로 그룹화 하는 과정으로, 하나의 클러스터에 속하는 데이터 객체들 간에는 서로 다른 클러스터 내의 객체들과는 구분되는 유사성을 갖게 된다[2].

클러스터링 방법은 크게 분할 (partitioning) 방법과 계층적(hierarchical) 방법으로 나눌 수 있다. 분할 방법은 범주 함수를 최적화 시키는 K개의 분할영역을 결정해 나가는 방법으로, Euclidean distance 측정법에 기반을 둔다. 계층적 방법은 통합 (agglomeratvie) 방법과 분리 (divisive) 방법으로 나눌 수 있다. 통합 방법은 처음에 각각의 데이터 객체를 하나의 클러스터로 설정 한 후 이들 쌍간의 거리를 기반으로 가장 가까운 클러스터들끼리 합병을 수행한다. 최종적으로 한 클러스터에 모든 데이터 객체들이 포함될 때까지 위의 과정을 반복한다. 분리 방법은 통합 방법과 반대로 위의 과정을 진행한다.

클러스터링 기법들은 통계학(statistics), 패턴인식(pattern recognition) 등의 분야에서 연구되어 왔으며, 현재는 데이터 마이닝 분야에서 이 기법을 응용하려는 연구가 활발히 진행되고 있다. 기존의 클러스터링 기법들은 주로 수치형 데이터 [4][7][8][11]와 범주형 데이터[9][12]들을 문제영역으로 다루어 왔다. 그러나, 실제로는 범주형 값들이 순서를 가지고 있는 데이터들이 존재하며, 기존의 기법들은 이러한 데이터들을 고려하지 않았다.

범주형 값들이 순서를 가지고 있는 데이터들에 대한 클러스터링은 행동에 의한 세분화(behavioral segmentation) 분야에 많은 응용 문제를 가지고 있다[5]. 예를 들면 웹 사용자를 세분화하는 문제에서, 웹 로그 파일을 이용하여 웹 사용자들을 클러스터링 하는 문제이다.

본 논문에서는 웹 로그나 소매점 거래 데이터 등과 같이 범주형 값들이 순서를 가지고 있는 시퀀스들을 클러스터링 하는 문제를 다룬다. 이를 위해서는 시퀀스들 간의 유사도를 구하는 것이 무엇보다 중요하다. 이를 위해, 새로운 유사도 척도와 이 척도를 이용한 계층적 클러스터링 방법을 제안한다.

2. 기존 연구

다양한 클러스터링 기법들에 대한 연구는 Jiawei et al.(2001)에 있고[2][3], 이중에서 범주형 속성들에 대한 연구는 Sudipto et al.(1999)과 Wang et al.(1999)에 있다[9][12]. Sudipto et al.(1999)과 Wang et al.(1999)는 단지 데이터들이 범주형 속성들로 이루어진 경우의 클러스터링 문제만을 다루고 있다.

범주형 값의 시퀀스에 대한 연구는 주로 빈발하는 순차 패턴을 찾는데 집중되어 왔다. 이 문제는 Rakesh et al.(1995)에서 처음으로 제안되었는데, 이 분야의 순차패턴을 탐사하는 문제는 시퀀스의 지지도가 사용자가 정의한 최소지지도보다 같거나 큰 시퀀스를 발견하는 것이다[6].

복잡한 객체들의 시퀀스를 클러스터링 하는 문제는 Alian et al.(1997)에서 제안되었다[1]. 이 논문은 클래스 계층을 사용하여 클러스터링을 수행하는데, 수치형 데이터로 표현되는 객체들(예를 들면 움직이는 객체들의 궤적)의 시퀀스를 클러스터링 하는데 적합하다.

Tadeusz et al.(2001)에서 범주형 값들이 순서를 가지고 있는 데이터들의 클러스터링을 제안하였다[10]. 이 논문은 빈발 패턴이 주어져 있다고 가정을 하고, 이 빈발 패턴을 하나 이상 포함한 데이터만을 대상으로 클러스터링을 수행한다. 그러나, 본 연구에서는 빈발 패턴에 상관없이 본 논문에서 제안하는 새로운 유사도 척도에 따라 모든 데이터를 대상으로 클러스터링을 진행한다.

3. 유사도 측정

본 연구에서 사용되는 용어와 제안하는 유사도 계산 방법을 설명한다.

정의 1. $I = \{ i_1, i_2, \cdots, i_j, \cdots, i_m \}$은 항목 i_j들의 집합이다.

정의 2. 시퀀스(sequence) S는 n 개의 항목들의 집합이고 $< x_1\ x_2\ \cdots\ x_j\ \cdots\ x_n >$으로 표시하고 여기서 x_j는 항목이다. 데이터베이스 D는 시퀀스들의 집합이다.

정의 3. 시퀀스의 크기는 그 시퀀스에 존재하는 항목들의 개수이며, 크기가 k인 시퀀스를 k-시퀀스라고 한다.

정의 4. 시퀀스 $S = < x_1\ x_2\ \cdots\ x_i\ x_j\ \cdots\ x_n >$에서 2 개의 항목집합으로 구성된 $x_i\ x_j$ $(i < j)$를 시퀀스 요소 e_i 라고 하며, 시퀀스 S에는 $\sum_{k=1}^{n-1} k$ 개의 시퀀스 요소가 존재한다. $E = \{ e_1, e_2, \cdots, e_i, \cdots, e_{n-1} \}$는 시퀀스 요소 e_i 들의 집합이다.

정의 5. 두 개의 시퀀스 $S_1 = < a_1\ a_2\ \cdots\ a_n >$과 $S_2 = < b_1\ b_2\ \cdots\ b_m >$ 의 시퀀스 요소 집합을 각각 $E_1 = \{ ea_1, ea_2, \cdots, ea_i, \cdots, ea_{n-1} \}$, $E_2 = \{ eb_1, eb_2, \cdots, eb_j, \cdots, eb_{m-1} \}$라고 하면, S_1, S_2 의 유사도 $sim(S_1, S_2)$는 다음과 같이 정의한다.

$$sim(S1,S2) = \sum_{ea_i \in E_1,\, eb_j \in E_2} \delta(ea_i, eb_j) \quad \cdots\cdots (1)$$

여기서, $\delta(ea_i, eb_j) \begin{cases} = 1 & \text{if } ea_i = eb_j \\ = 0 & \text{otherwise} \end{cases}$

예) 두 개의 시퀀스 $S_1 = <$ ABDE $>$, $S_2 = <$ ACDEG $>$가 있다. S_1 과 S_2 는 각각 4-시퀀스와 5-시퀀스이며, 각각의 시퀀스 요소 집합은 각각 $E_1 = \{$ AB, AD, AE, BD, BE, DE $\}$ 과 $E_2 = \{$ AC, AD, AE, AG, CD, CE, CG, DE, DG, EG $\}$ 이다. 두 시퀀스의 유사도 $sim(S_1, S_2)$는 3이다.

정의 6. 클러스터 $C_1 = \{ S_1, S_2, \cdots, S_i, \cdots, S_n \}$과 클러스터 $C_2 = \{ S_1, S_2, \cdots, S_j, \cdots, S_m \}$의 유사도 $sim(C_1, C_2)$는 다음과 같이 정의한다.

$$sim(C_1, C_2) = \frac{1}{|C_1|\,|C_2|} \sum_{S_i \in C_1,\, S_j \in C_2} sim(S_i, S_j) \quad \cdots\cdots (2)$$

여기서, $|C_1|, |C_2|$ 는 각각 클러스터 C_1, C_2 에 있는 시퀀스의 총 개수

예) 네 개의 시퀀스 $S_1 = <$ ABDE $>$, $S_2 = <$ ACDEG $>$, $S_3 = <$ ABG $>$, $S_4 = <$ AEG $>$ 로 이루어진 두 개의 클러스터 $C_1 = \{ S_1, S_2 \}$, $C_2 = \{ S_3, S_4 \}$ 가 있다. 두 클러스터 C_1, C_2 의 유사도 $sim(C_1, C_2)$는 다음과 같다.

$$\begin{aligned} sim(C1, C2) &= \frac{1}{2 \cdot 2} \{sim(S1,S3)+sim(S1,S4) \\ &\quad +sim(S2,S3)+sim(S2,S4)\} \\ &= \frac{1}{2 \cdot 2} \{1+1+1+3\} = 1.5 \end{aligned}$$

4. 클러스터링 알고리듬

4.1 계층적 클러스터링 알고리듬

본 연구에서 제안하는 클러스터링 알고리듬의 개요는 <그림 1>과 같다.

```
Step 0. 초기화
    for all i
        Ci ← Si ∈ D

Step 1. 유사도 계산
    for each Ci, Cj ∈ D
        Compute sim(Ci, Cj)

Step 2. 병합
    Cnew ← Merge (Ci, Cj)
            for max_sim(Ci, Cj)

Step 3. 유사도 갱신
    Update sim(Cnew, Ci)

Step 4. 조건 검사
    If ( | Ci | > n ) And
       ( exist Ci, Cj ∈ D such that
          sim(Ci, Cj) ≥ min_sim )
    Then Goto Step 2.
    Else Goto Step 5.

Step 5. 종료
    끝낸다
```

<그림 1> 클러스터링 알고리듬

Step 0은 초기화 단계로서 데이터베이스 D를 액세스하여 각각의 시퀀스를 하나의 클러스터로 설정한다.

Step 1은 유사도 계산 단계로서 각 클러스터간의 유사도를 계산한다.

Step 2는 클러스터 합병 단계로서 유사도가 가장 높은 두 개의 클러스터를 합병한다.

Step 3은 유사도 갱신 단계로서 Step 2에서 합병된 클러스터와 나머지 클러스터간의 유사도를 갱신한다.

Step 4는 조건 검사 단계로서 클러스터의 개수가 지정된 클러스터 개수보다 크고 두 클러스터간의 유사도중 최소 유사도 이상의 것이 존재하면 Step 2로 간다. 그렇지 않으면 Step 5로 간다.

마지막으로, Step 5는 종료단계로서 알고리듬을 끝낸다.

웹 사용자 세분화 문제를 예로 들어 설명하겠다. <그림 2>와 같은 웹 로그 파일이 주어져 있다. (예 : 사용자 u1은 사이트 A를 방문한 후, 차례대로 B, C, F, Z를 방문함) 웹 사용자들을 웹 로그 파일을 기초로 2개의 그룹으로 클러스터링 하는 문제이다. 즉, 웹 사용자의 사이트 방문 기록을 다른 사용자의 방문 기록과 비교하여 클러스터링을 하는 문제이다.

```
u1 : A→B→C→F→Z
u2 : A→C→F→H
u3 : B→E→G→I
u4 : B→G→I
u5 : C→B→A→F
u6 : Z→C→B→A
```

<그림 2> 웹 사용자들의 사이트 방문기록

<그림 1>의 클러스터링 알고리듬에서 보면, Step 0에서 각 시퀀스들을 하나의 클러스터로 할당하여 <그림 3>과 같이 모두 6개의 클러스터를 생성한다.

c1	u1 : A→B→C→F→Z
c2	u2 : A→C→F→H
c3	u3 : B→E→G→I
c4	u4 : B→G→I
c5	u5 : C→B→A→F
c6	u6 : Z→C→B→A

<그림 3> 최초의 클러스터들

Step 1에서 각 클러스터들간의 유사도를 계산하고, Step 2에서 유사도가 가장 높은 u1과 u2를 합병한다. 그러면 <그림 4>와 같이 모두 5개의 클러스터가 생성된다.

c1	u1 : A→B→C→F→Z u2 : A→C→F→H
c2	u3 : B→E→G→I
c3	u4 : B→G→I
c4	C→B→A→F
c5	Z→C→B→A

<그림 4> 첫 번째 합병 후의 클러스터들

이후, Step 3에서 유사도를 갱신한 후, 현재의 클러스터 개수가 지정된 클러스터 개수 2보다 크고 최소 유사도 이상의 클러스터가 있으면 앞의 과정을 반복 수행한다.

4.2 유사도를 효율적으로 계산하는 알고리듬

본 알고리듬에서 가장 많은 계산량을 필요로 하는 부분은 유사도 계산 및 갱신 단계이다. 따라서, 이들 유사도 계산을 효율적으로 수행하기 위하여 <그림 5>와 같은 유사도 계산 알고리듬을 제안한다. <그림 5>는 두 개의 시퀀스 $S_1 = < a_1\ a_2 \cdots\ a_n >$과 $S_2 = < b_1\ b_2 \cdots\ b_m >$의 유사도를 계산하는 알고리듬이다.

```
step 0. 초기화
        Sa = { }, Sb = { }, count=0

Step 1. Sa 생성
        for i=1 to n {
           for j=1 to m {
             if ai = bj
                insert(Sa, ai)
           }
         }

Step 2. Sb 생성
        for j=1 to m {
           for i=1 to n {
             if bj = ai
                insert(Sb, bj)
           }
        }

Step 3. Sa, Sb 간의 유사도 계산
        for i=1 to n'
          for j=1 to m' {
            if eai = ebj
               count=count+1
          }
        }

Step 4. 종료
        sim(S1, S2) = count
```

<그림 5> 효율적인 유사도 계산 알고리듬 : $sim(S_1, S_2)$

Step 0에서 S_a, S_b, count 값을 초기화한다.

Step 1에서는 시퀀스 S_1의 항목들을 S_2의 항목들과 비교하여 공통 항목들로만 구성된 시퀀스 S_a를 만든다.

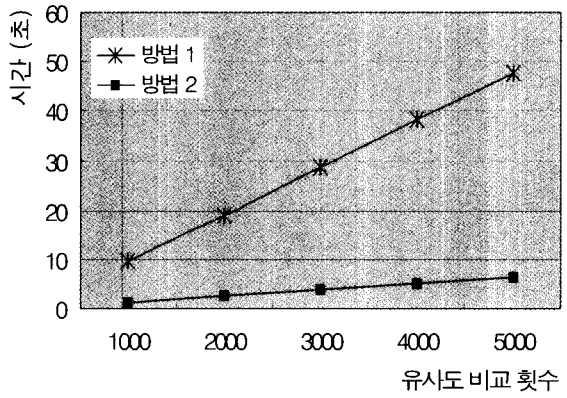
Step 2에서는 시퀀스 S_2의 항목들을 S_1의 항목들과 비교하여 공통 항목들로만 구성된 시퀀스 S_b를 만든다.

Step 3에서는 S_a과 S_b의 공통 시퀀스 요소들의 개수를 구하고 Step 4에서 알고리듬을 종료한다.

예를 들어, $S_1 = <$ ABCFZ $>$, $S_2 = <$ ACFH $>$의 유사도 계산 과정을 주어진 시퀀스로부터 직접 계산하면 S_1의 시퀀스 요소 {AB, AC, AF, AZ, BC, BF, BZ, CF, CZ, FZ}를 S_2 의 시퀀스 요소 {AC, AF, AH, CF, CH, FH}와 비교하여 유사도를 계산한다. (정의 5 참조) 그러나, <그림 5>의 알고리듬을 사용하면, Step 1에서 S_a = <ACF>를 구하고, Step 2에서 S_b = <ACF> 구한다. 그후, Step 3에서 S_a의 시퀀스 요소들인 {AC, AF, CF}와 S_b의 시퀀스 요소들인 {AC, AF, CF}를 이용하여 유사도를 계산하기 때문에, S_1, S_2로부터 직접 유사도를 계산하는 방법보다 효율적으로 유사도를 계산할 수 있다.

5. 실험결과

본 연구에서 제안하는 클러스터링 알고리듬의 성능을 평가하기 위해, 클러스터링 알고리듬 중 가장 많은 계산량을 필요로 하는 유사도 계산 부분에 대한 실험을 수행하였으며, 결과는 <그림 6>에 있다.



<그림 6> 실험결과

본 실험은 인텔 550MHz 사양의 펜티엄 Ⅲ 컴퓨터에서 C++ 언어로 코딩하여 수행하였고, 시퀀스의 크기가 12~15인 데이터를 대상으로 하였다.

<그림 6>에서 방법 1은 주어진 데이터로부터 직접 유

사도를 계산하는 방법이고, 방법 2는 본 연구에서 제안하는 효율적인 유사도 계산 알고리듬인 <그림 5>를 이용한 방법이다.

<그림 6>에서 보는 바와 같이 본 연구에서 제안하는 유사도 계산 알고리듬의 수행시간이 방법 1보다 훨씬 적음을 알 수 있다. 또한, 데이터들의 수가 늘어남에 따라 유사도 비교 횟수가 증가할수록, 본 연구에서 제안하는 유사도 계산 알고리듬과 방법 1과의 실행 시간 차이는 커짐을 알 수 있다.

6. 결 론

본 논문에서는 범주형 값들이 순서를 가지고 있는 데이터들의 클러스터링 문제를 연구하였다. 본 문제를 풀기 위하여 시퀀스 요소를 이용한 새로운 유사도 척도를 제안하였고, 이 척도를 이용한 계층적 클러스터링 알고리듬을 제안하였다. 또한 클러스터링 알고리듬에서 가장 많은 계산량을 필요로 하는 유사도 계산 과정을 효율적으로 수행할 수 있는 유사도 계산 알고리듬도 제안하였고, 실험을 통하여 성능의 우수함을 보였다.

본 논문의 클러스터링 알고리듬은 기존의 방법과 달리 빈발 패턴이 주어져 있지 않아도 되며, 모든 입력 데이터들을 대상으로 지정된 클러스터 개수나 최소 유사도 이하의 클러스터가 없을 때까지 클러스터링을 수행한다.

향후 과제로는, 다양한 데이터셋들을 대상으로 제안하는 클러스터링 방법에 대한 결과분석과 시퀀스들간의 유사도 계산시 다양한 조건을 고려할 수 있는 유사도 계산 방법에 대한 연구가 필요하겠다.

참고문헌

[1] Alian K.; "Clustering Sequences of Complex Objects", Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, 1997

[2] Jiawei H. and Micheline K.; Data Mining : Concepts and Techniques, Morgan kaufmann Publishers, pp335-393, 2001

[3] Jiawei H., Micheline K., and Anthony K. H. Tung; "Spatial Clustering Methods in Data Mining : A Survey", H. J. Miller and J. Han (eds.), Geographic Data Mining and Knowledge Discovery, NY : Taylor and Francis, 2001.

[4] Martin E., Hans-Peter K., Jorg S., and Xiaowei X.;"A Dennsity-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", KDD, 1996.

[5] Perkowitz M. and Etzioni O.; "Towards Adaptive Web Sites : Conceptual Framework and Case Study", Computer Networks 31, Proceedings of the 8th International WWW Conference, 1999

[6] Rakesh A. and Ramakrishnan S.; "Mining Sequential Patterns", Proceedings of the 11th International Conference on Data Engineering, 1995

[7] Raymond T. Ng and Jiawei H.; "Efficient and Effective Clustering Method for Spatial Data Mining", VLDB 1994.

[8] Sudipto G., Rajeev R., and Kyuseok S.; "CURE : An Efficient Clustering Algorithm for Large Databases", SIGMOD98, 1998

[9] Sudipto G., Rajeev R., and Kyuseok S.; "ROCK : A Robust Clustering Algorithm for Categorical Attributes", IEEE99, 1999

[10] Tadeusz M., Marek W.; and Maciej Z. : "Scalable Hierarchical Clustering Method for Sequences of Categorical Values", Proc. of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'01), Kowloon, Hong Kong, 2001

[11] Tian Z., Raghu R., and Miron L.; "BIRCH : An Efficient Data Clustering Method for Very Large Databases", ACM SIGMOD96, 1996

[12] Wang K., Xu C., and Liu B.; "Clustering Transactions Using Large Items", Proceedings of the '99 ACM, 1999