퍼지 및 지능시스템학회 논문지 2004, Vol. 14, No. 2, pp. 136-141

범주형 시퀀스들에 대한 확장성 있는 클러스터링 방법

A Scalable Clustering Method for Categorical Sequences

오승준 · 김재련
Seung-Joon Oh and Jae-Yearn Kim

한양대학교 산업공학과

요 약

소매점 거래 데이터와 단백질 시퀀스, 웹 로그 등과 같은 상업적이거나 과학적인 데이터의 폭발적인 증가를 볼 수 있다. 이런 데이터들은 순서적인 면을 가지고 있는 시퀀스 데이터들이다. 그러나, 순서적인 면을 고려한 클러스터링 알고리듬은 소수이다. 따라서, 본 연구에서는 시퀀스 데이터들을 클러스터링 하는 방법을 연구한다. 시퀀스들 간의 유사도를 계산하기 위한 새로운 유사도를 제안한다. 또한, 유사도를 효율적으로 계산하기 위한 방법과 클러스터링 방법도 제안한다. 계층적 클러스터링 알고리듬은 높은 계산량을 가지고 있기에, 새로운 클러스터링 방법이 요구된다. 그러므로, 본 연구에서는 샘플링과 k-nn 방법을 이용한 확장성 있는 클러스터링 방법을 제안한다. 실제 데이터 셋과 합성 데이터 셋을 이용하여, 본 연구에서 제안하는 방법이 기존 방법보다 성능이 우수함을 보여준다.

Abstract

There has been enormous growth in the amount of commercial and scientific data, such as retail transactions, protein sequences, and web-logs. Such datasets consist of sequence data that have an inherent sequential nature. However, few clustering algorithms consider sequentiality. In this paper, we study how to cluster sequence datasets. We propose a new similarity measure to compute the similarity between two sequences. We also present an efficient method for determining the similarity measure and develop a clustering algorithm. Due to the high computational complexity of hierarchical clustering algorithms for clustering large datasets, a new clustering method is required. Therefore, we propose a new scalable clustering method using sampling and a k-nearest-neighbor method. Using a real dataset and a synthetic dataset, we show that the quality of clusters generated by our proposed approach is better than that of clusters produced by traditional algorithms.

Key words : Data Mining, Clustering, Sampling, K-Nearest Neighbor

1. 서 론

클러스터링(Clustering)이란 물리적 혹은 추상적 객체들을 서로 비슷한 객체들의 집합으로 그룹화 하는 과정으로, 하나의 클러스터에 속하는 객체들 간에는 서로 다른 클러스터 내의 객체들과는 구분되는 유사성을 갖게 된다[1]. 클러스터링 기법들은 통계학(statistics), 패턴인식(pattern recognition) 등의 분야에서 연구되어 왔으며, 현재는 데이터 마이닝 분야에서 이 기법을 응용하려는 연구가 활발히 진행되고 있다.

최근에는 상업적이거나 과학적인 데이터의 폭발적인 증가를 볼 수 있다. 이들 중 웹 로그나 단백질 시퀀스, 소매점 거래 데이터 등과 같은 분야의 데이터들은 순서적인 면을 가지고 있는 시퀀스 데이터(또는 시퀀스)들이다.

항목들 간에 순서가 존재하는 시퀀스들을 클러스터링 하는 것은 많은 면에서 유용하다. 예를 들면, 웹 사용자들의 사이트 방문기록을 보관한 웹 로그 파일들을 이용하여 웹 사용자들을 클러스터링 하는 것은 서로 다른 웹 사용자 그룹들을 발견하는데 도움을 준다[2]. 또한, 비슷한 구조를 공유하는 단백질 시퀀스들끼리 그룹화 하는 것은 비슷한 기능을 갖는 시퀀스들을 찾는데 도움을 준다.

그러나, 이러한 시퀀스 데이터들에 대한 클러스터링 연구는 소수에 불과하다. 그래서, 본 연구에서는 웹 로그나 단백질 시퀀스, 소매점 거래 데이터 등과 같이 항목들 사이에 순서가 존재하는 시퀀스들을 클러스터링 하는 문제를 다룬다.

시퀀스들을 클러스터링 하기 위해서는 시퀀스들 간의 유사도를 구하는 것이 무엇보다 중요하다. 시퀀스들 간의 유사도가 효율적으로 계산되어진다면, 계층적 클러스터링 알고리듬을 이용하여 시퀀스들을 클러스터링 할 수 있게 된다.

시퀀스들 사이의 유사도를 계산하는 방법에는 edit distance 방법[3][4][5]과 sequence alignment 방법[3][6]이 있으며, 수학적인 관점에서 보면, edit distance 방법과 sequence alignment 방법은 동일하다[3]. edit distance 방법은 유사도 계산시 시퀀스 전체를 고려하기 때문에, 때로는 중요한 특성을 나타내는 서브 시퀀스들을 고려하지 못하며, 다수의 edit operations 조합이 가능하다. sequence alignment 방법은 scoring scheme에 의존적이며, 항목 값들의 종류가 적은 경우에만 효율적이다. 그래서 본 연구에서는 이들 기존 방법들의 단점을 개선한 새로운 유사도 계산 방법을 제안한다.

또한, 대규모의 데이터들을 클러스터링 하는 경우에는 계

접수일자 : 2004년 2월 23일
완료일자 : 2004년 3월 18일

층적 클러스터링 알고리듬은 많은 계산량 때문에 적용이 불가능하므로, 새로운 클러스터링 방법이 요구된다. 그래서, 본 연구에서는 샘플링과 k-nearest neighbor(k-nn)방법을 이용하여 대규모의 데이터들에도 적용이 가능한 새로운 클러스터링 방법을 제안한다.

2. 기존 연구

기존의 클러스터링 기법들은 주로 수치형 값들의 데이터[7][8]와 범주형 값들의 데이터[9][10]들만을 문제영역으로 다루어 왔다.

시퀀스에 대한 연구는 주로 빈발하는 순차 패턴을 찾는데 집중되어 왔다. 이 문제는 Agrawal and Srikant [11]에서 처음으로 제안되었는데, 이 분야의 순차패턴을 탐사하는 문제는 시퀀스의 지지도가 사용자가 정의한 최소지지도보다 큰 시퀀스를 발견하는 것이다. Joshi et al. [12]에서는 순차 패턴을 일반화 시켜 표현하는 방법을 다루었다.

시퀀스들에 대한 클러스터링 연구로는 다음의 세 가지 연구가 있다. Morzy et al. [13]은 빈발패턴이 주어져 있다고 가정을 하고, 이 빈발 패턴을 하나 이상 포함한 시퀀스들만을 대상으로 클러스터링을 수행한다. Hay et al. [14]은 시퀀스들 사이의 유사도로 edit distance 방법을 사용하여 클러스터링을 수행하고, Wang and Zaiane [15]는 sequence alignment 방법을 이용하여 클러스터링을 수행한다. 그러나, 본 연구에서는 Morzy et al. [13]와 달리 빈발패턴에 상관없이 모든 시퀀스들을 대상으로 클러스터링을 수행하고, Hay et al. [14]나 Wang and Zaiane [15]에서 사용한 유사도 계산 방법과 다른 새로운 유사도를 사용하여 시퀀스들을 클러스터링 한다.

계층적 클러스터링 방법의 단점을 보완하기 위하여 샘플링을 이용한 연구로는 S. Guha et. al. [7][9]가 있다. 그러나, 이들은 시퀀스 데이터가 아니라 범주형이나 수치형 값들로만 이루어진 데이터들을 대상으로 클러스터링을 수행한다.

3. 시퀀스들 간의 유사도

3.1 유사도 측정

클러스터링을 수행하기 전에 먼저 시퀀스들 간의 유사도(혹은 거리)를 측정해야 한다. 일반적으로 두 시퀀스들 간의 유사도는 공통 항목이 많을수록, 또한 항목들의 순서가 동일할수록 높다고 할 수 있다. 따라서, 이 두 가지 요소를 동시에 고려하기 위해서는 두 시퀀스 사이에 동일 서브셋들이 얼마나 많이 존재하느냐를 고려한다. 본 연구에서는 동일 서브셋들을 찾기 위해 순서를 가지는 두 항목 쌍들을 이용한다. 즉, 두 시퀀스들 사이에 동일 항목 쌍들이 많을수록 유사도가 높게 나오는 성질을 이용한다.

3.2 유사도 계산 방법

데이터베이스 D는 시퀀스들의 집합이고, 시퀀스 S는 n개의 항목들의 모임이며 $\langle x_1 x_2 \cdots x_i \cdots x_j \cdots x_n \rangle$으로 표시하고 여기서 x_i는 범주형 값을 가지는 항목이다. S의 크기는 S에 있는 항목들의 개수이며, $|S|$로 나타낸다. 시퀀스 S에서 순서를 가지는 2개의 항목들로 구성된 $x_i x_j (i < j)$를 시퀀스 요소 e_k 라고 하며, e_k들의 모임을 $E=(e_1, e_2, \ldots, e_k, \ldots)$라 한다. E의 크기는 E에 있는 요소들의 개수이며, $|E|$로 나타낸다.

시퀀스내의 항목들뿐만 아니라 항목들 간의 순서도 고려를 해서 식(1)과 같이 유사도 계산 방법을 제안한다.

정의 3.1 두 시퀀스 S_1과 S_2의 시퀀스 요소들의 모임을 각각 E_1, E_2라고 하면, S_1, S_2의 유사도 $sim(S_1, S_2)$는 다음과 같이 정의한다.

$$sim(S_1, S_2) = \frac{|E_1 \cap E_2|}{\frac{|E_1| + |E_2|}{2}} \quad (1)$$

여기서, $|E_1 \cap E_2|$는 E_1과 E_2의 공통 요소들의 개수이며, E_1과 E_2사이에 공통 항목들이 많을수록 유사도는 높고, 이 값을 $(|E_1| + |E_2|)/2$로 나누는 것은 유사도를 0과 1사이의 값을 갖도록 하기 위해서이다.

예제 3.1 두 시퀀스 $S_1 = \langle A\ B\ C\ A \rangle$, $S_2 = \langle A\ C\ D\ A\ C \rangle$에서 시퀀스 요소들의 모임은 각각 E_1 = (AB, AC, AA, BC, BA, CA)과 E_2 = (AC, AD, AA, AC, CD, CA, CC, DA, DC, AC) 이며, $|E_1| = 6$, $|E_2| = 10$, $E_1 \cap E_2$ = (AC, AA, CA), $|E_1 \cap E_2| = 3$이다. 따라서, 두 시퀀스의 유사도 $sim(S_1, S_2)$는 3/8이다.

제안하는 유사도 계산 방법을 기존 방법들인 edit distance와 sequence alignment 방법과 비교해 보자. 먼저, 다음과 같은 세 시퀀스 $S_1 = \langle A\ B\ C\ D\ E\ F \rangle$, $S_2 = \langle D\ E\ F\ A\ B\ C \rangle$, $S_3 = \langle D\ E\ F\ G\ H\ I \rangle$가 있다. $sim(S_1, S_2)$, $sim(S_1, S_3)$를 edit distance 방법을 이용하여 구하면, $sim(S_1, S_2) = 6$, $sim(S_1, S_3) = 6$이다. 그러나, 제안하는 유사도 방법을 이용하면, $sim(S_1, S_2) = 6/15$, $sim(S_1, S_3) = 3/15$이다. 즉, 일부 서브시퀀스들의 자리가 바뀔 경우 (블럭 operations), 제안하는 유사도 방법은 edit distance 방법보다 효율적으로 유사도를 계산할 수 있다. 또한, 제안하는 유사도 방법은 시퀀스 요소를 이용하여 두 시퀀스 사이의 유사도를 계산하므로, edit distance 방법처럼 다수의 edit operation 조합이 생성되지 않는다.

sequence alignment 방법은 scoring scheme에 의존적이며, 항목 값들의 종류가 적을 경우에 효율적으로 유사도를 계산하는데 비해, 제안하는 유사도 방법은 시퀀스 요소를 이용함으로써, 항목 값들의 종류가 많을 경우에도 유사도를 효율적으로 계산할 수 있다. (성질 3.1 참조)

3.3 효율적인 유사도 계산 방법

식(1)에서 보면, 시퀀스들 간의 유사도를 계산하는데 두 시퀀스들 간의 공통된 시퀀스 요소들의 개수를 구하는 것이 중요하다. 따라서, 공통된 시퀀스 요소들의 개수를 효율적으로 구하는 성질 3.1을 이용하면 시퀀스들 간의 유사도를 효율적으로 계산할 수 있다.

성질 3.1 두 시퀀스 $S_1 = \langle a_1 \cdots a_i \cdots c_k \cdots a_n \rangle$, $S_2 = \langle b_1 \cdots b_j \cdots c_l \cdots b_m \rangle$가 있다. c_k, c_l 항목들은 S_1, S_2 에 공통으로 있는 항목들이며, a_i, b_j 항목들은 각각 S_1, S_2 에만 있는 항목들이다. c_k 항목들로만 이루어진 시퀀스를 S_3라 하고, c_l 항목들로만 이루어진 시퀀스를 S_4라 하자. 또한, E_1, E_2, E_3, E_4 를 각각 S_1, S_2, S_3, S_4 의 시퀀스 요소들의 모임이라고 하면, S_1, S_2 의 유사도 $sim(S_1, S_2)$는 다음과 같이 계산된다.

$$sim(S_1, S_2) = \frac{|E_3 \cap E_4|}{\frac{|E_1| + |E_2|}{2}} \quad (2)$$

(증명) 식(1)에 의하여 $sim(S_1, S_2) = \frac{|E_1 \cap E_2|}{\frac{|E_1| + |E_2|}{2}}$

이다. 여기서 $|E_1 \cap E_2| = |E_3 \cap E_4|$ 이다. 왜냐하면, $|E_1 \cap E_2|$ 는 S_1과 S_2에 공통으로 존재하는 시퀀스 요소들의 개수이기 때문에 S_1, S_2에서 서로 자신들에게만 존재하는 항목들을 제외한 S_3, S_4의 공통 시퀀스 요소들의 개수인 $|E_3 \cap E_4|$ 를 구하여도 마찬가지의 결과를 얻는다.

그러므로, $sim(S_1, S_2) = \frac{|E_3 \cap E_4|}{\frac{|E_1| + |E_2|}{2}}$

예제 3.2 두 시퀀스 S_1=<A B C F D A>, S_2=<A F C H>의 유사도 $sim(S_1, S_2)$를 계산해 보자. S_1, S_2 로부터 직접 유사도를 계산하는 경우에는 S_1의 시퀀스 요소들인 (AB, AC, AF, AD, AA, BC, BF, BD, BA, CF, CD, CA, FD, FA, DA)를 S_2의 시퀀스 요소들인 (AF, AC, AH, FC, FH, CH)와 비교하여 유사도를 계산한다.

그러나, 성질 3.1을 사용하기 위해, S_1, S_2의 공통 항목들로만 구성된 S_3=<A C F A>과 S_4=<A F C>를 구한다. S_3는 다음과 같이 생성을 한다. S_1의 모든 항목들을 차례대로 S_2의 항목들과 비교하여 동일 항목이 존재하면 S_3에 추가시키면서, S_3을 생성한다. 마찬가지 방법으로 S_2의 모든 항목들을 차례대로 S_1의 항목들과 비교하여 S_4를 생성한다. 그 후, S_3의 시퀀스 요소들인 (AC, AF, AA, CF, CA, FA)와 S_4의 시퀀스 요소들인 (AF, AC, FC)를 이용하여 유사도를 계산한다. 그러므로, S_1과 S_2로부터 직접 유사도를 계산하는 것보다 훨씬 효율적으로 유사도를 계산할 수 있다.

4. 확장성 있는 클러스터링 방법

4.1 클러스터링 방법의 개요

클러스터링 방법의 전체 개요는 그림 1과 같다.

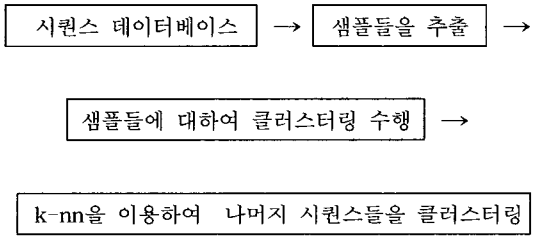

그림 1. 클러스터링 방법의 개요
Fig 1. Overview of the clustering method

시퀀스 데이터베이스로부터 샘플들을 추출한 후에, 샘플들에 대하여 계층적 클러스터링 알고리듬을 수행한다. 다음으로, 샘플들만으로 이루어진 클러스터들과 나머지 시퀀스들을 k-nearest neighbor (k-nn) 방법을 이용하여 클러스터링 한다.

4.2 랜덤 샘플링 단계

데이터베이스의 크기가 클 경우에 랜덤 샘플링을 이용하면 고려해야 할 데이터의 크기를 줄일 수 있으며, 이로 인해 클러스터링 실행시간을 단축시킬 수 있다. 또한, 적당한 양의 샘플들을 이용하면, 클러스터링의 질을 떨어뜨리지 않을 수 있으며, 아웃라이어들을 필터링함으로써 클러스터링의 질을 향상시킬 수도 있다.[7][9] 본 연구에서도 대규모의 데이터셋을 효율적으로 처리하기 위해 랜덤 샘플링을 이용하여 샘플들을 추출한 후, 이들을 이용한다.

4.3 샘플들을 클러스터링 하는 단계

본 연구에서는 샘플들을 클러스터링하기 위해서 통합 방법의 계층적 클러스터링 알고리듬을 사용한다. n개의 시퀀스들을 클러스터링 하는 문제를 생각해 보자. 처음에는 $n \times (n-1)/2$개의 클러스터간 합병을 고려할 수 있는데, 이중에서 합병을 했을 경우 가장 높은 평가함수 값을 주는 두 개의 클러스터를 합병한다. 1번째 합병 후에는 $(n-1) \times (n-1-1)/2$개의 클러스터간 합병을 고려하며, 이중에서 가장 높은 평가함수 값을 주는 두 개의 클러스터를 합병한다. 최종적으로는 주어진 개수의 클러스터가 남을 때까지 위의 과정을 반복한다.

본 연구에서는 평가함수로 식(3)를 사용한다.

$$\text{Maximize } Cf = \sum_{r=1}^{k} \frac{1}{n_r} \sum_{i,j \in C_r} sim(i, j) \quad (3)$$

여기서, n_r은 C_r 내의 시퀀스들 개수, k는 클러스터 개수

본 연구에서 제안하는 클러스터링 알고리듬의 단계는 그림 2와 같다. 여기서 C_i와 S_i는 각각 i번째 클러스터와 시퀀스이며, $|C_i|$ 는 현재 단계의 클러스터 개수이다.

```
Step 0. 초기화
    for all i
        Ci ← Si ∈ D
Step 1. 평가함수의 값 계산
    for each Ci, Cj ∈ D
        Compute the criterion function (식(3))
Step 2. 합병
    Cnew ← Merge (Ci, Cj) for
            the max value of the criterion function
Step 3. 조건 검사
    If ( |Ci| > n ) Then Goto Step 1.
    Else Goto Step 4.
Step 4. 종료
    끝낸다.
```

그림 2. 계층적 클러스터링 알고리듬
Fig 2. Proposed hierarchical clustering algorithm

Step 0은 초기화 단계로서 데이터베이스 D를 액세스하여 각각의 시퀀스를 하나의 클러스터로 설정한다. Step 1은 두 클러스터가 합병이 될 경우의 평가함수 (3)의 값을 구하는 단계로, 현재 n개의 클러스터가 있다고 하면, $n \times (n-1)/2$개의 평가함수 값을 계산한다. Step 2는 합병 단계로서, Step 1에서 계산한 평가함수 값들 중 가장 큰 값을 주는 두 개의 클러스터를 합병한다. Step 3은 조건 검사 단계로서 클러스터의 개수가 지정된 클러스터 개수보다 크면 Step 1으로 간다 그렇지 않으면 Step 4로 간다. 마지막으로, Step 4는 종료단계로서 알고리듬을 끝낸다.

4.4 나머지 시퀀스들을 클러스터링 하는 단계

나머지 시퀀스들을 클러스터링 하기 위해서, 분류 기법으로 사용되고 있는 k-nearest neighbor(k-nn)을 이용한 새로운 방법을 제안한다. 본 단계의 알고리듬은 다음과 같다.

Step 0. 클러스터링 하고자 하는 시퀀스 S_i와 클러스터들에 속한 모든 시퀀스들과의 유사도를 계산한다.
Step 1. 시퀀스 S_i에 대하여 유사도가 가장 높은 순서대로 k개의 이웃 시퀀스들을 구한다.
Step 2. k개의 이웃 시퀀스들이 가장 많이 속해 있는 클러스터를 구한다.
Step 3. 단계 2에서 구한 클러스터에 시퀀스 S_i를 할당한다.
Step 4. 클러스터링 하고자 하는 시퀀스가 남아 있으면, Step 0으로 가고, 아니면 알고리듬을 끝낸다.

그림 3. 나머지 시퀀스들을 클러스터링 하는 알고리듬
Fig 3. Proposed clustering algorithm for the remaining sequences

Step 0에서는 시퀀스 S_i와 클러스터들에 속한 모든 시퀀스들 간의 유사도를 구한다. 이때, 시퀀스들 간의 유사도는 정의 3.1을 사용한다. Step 1에서는 시퀀스 S_i에 대하여 유사도가 가장 높은 순서대로 k 이웃 시퀀스들을 구한다. Step 2에서는 Step 1에서 구한 k개의 이웃 시퀀스들이 가장 많이 속해 있는 클러스터를 구한다. Step 3에서는 시퀀스 S_i를 Step 2에서 구한 클러스터에 할당한다. Step 4에서는 클러스터링 하고자 하는 시퀀스가 남아 있는지를 검사하여, 시퀀스가 남아 있으면 Step 0으로 가고, 그렇지 않으면 알고리듬을 끝낸다.

예제 4.1 9개의 샘플 시퀀스들(S_1, S_2, ... , S_9)을 3개의 클러스터로 클러스터링 한 결과가 그림 4와 같다

CL_1	CL_2	CL_3
S_1=<A B C J>	S_2=<E J F K>	S_4=<G J K I>
S_3=<A C J K>	S_5=<J F D K>	S_6=<H I J K>
S_7=<A B J>	S_8=<D J K F E>	S_9=<G H I>

그림 4. 샘플 시퀀스들로 구성된 세 개의 클러스터들
Fig 4. Three clusters consisting of sampled sequences

여기서, S_{10}=<C J K F>와 S_{11}=<G H I J>을 3-nn을 이용하여 분류하면 다음과 같다. 먼저 S_{10}과 S_1, S_2, ... S_9사이의 유사도를 계산한다. 다음으로, S_{10}과 유사도가 가장 높은 3개의 이웃 시퀀스들을 구하면 S_3, S_8, S_2이다. 여기서 S_3은 CL_1에, S_8와 S_2는 CL_2에 속한다. 그러므로, S_{10}은 CL_2에 할당된다. S_{11}에 대한 3개의 이웃 시퀀스들은 S_9, S_6, S_4이고, 이들은 모두 CL_3에 속해 있으므로, S_{11}은 CL_3에 할당된다.

예제 4.1에서 S_{10}을 기존 방법대로 클러스터 할당을 해 보자. 기존 방법들(S. Guha et. al. [7][9])에서는 유사도가 가장 높은 시퀀스를 찾아서, 그 시퀀스가 속해 있는 클러스터로 시퀀스를 할당하게 된다. 즉, S_{10}은 유사도가 가장 높은 S_3가 속해있는 CL_1에 할당이 된다. 그러나, 예제 4.1에서 보는 바와 같이, S_{10}은 S_3을 제외하고는 CL_1에 있는 시퀀스들보다는 CL_2에 있는 시퀀스들과 유사도가 높다. 그러므로, 본 연구에서는 기존 방법처럼 단순히 유사도가 가장 높은 하나의 데이터만을 이용하는 것이 아니라 k-nn을 이용함으로써 합리적인 클러스터 할당을 수행할 수 있다.

5. 실험결과

본 연구에서 제안하는 방법을 기존 방법들과 비교 평가하기 위해, splice 데이터셋과 합성 데이터셋으로 실험을 하였다. 본 실험은 인텔 2.4 GHz 사양의 펜티엄 IV 컴퓨터에서 C++ 언어로 코딩을 하여 수행하였다.

5.1 splice 데이터셋

splice 데이터셋은 UCI KDD 아카이브에 포함되어 있는 데이터셋이다[16]. 이 데이터셋은 60개의 항목을 가진 뉴클레오타이드(nucleotide) 시퀀스들을 포함하고 있으며, 각각의 시퀀스들은 엑손/인트론 경계 (exon/intron, EI라 부름)나 인트론/엑손 경계 (intron/exon, IE라 부름)에 속하는 클래스 레이블을 가진다. EI에 속하는 시퀀스들이 767개이며, IE에 속하는 시퀀스들이 768개이다.

splice 데이터셋을 3가지 클러스터링 알고리듬으로 실험을 수행하였다. 알고리듬 1은 시퀀스들 간의 유사도로 Hay et al. [14]처럼 edit distance 방법을 이용했으며, 본 연구에서 제안하는 계층적 클러스터링 알고리듬을 사용하여 클러스터링을 수행하였다. 알고리듬 2는 시퀀스들 간의 유사도로 알고리듬 1과 동일한 방법을 사용하였으며, 최장거리법을 이용한 계층적 클러스터링 방법을 사용하였다.

splice 데이터셋을 알고리듬 1, 2 와 제안하는 알고리듬 모두 2개의 클러스터로 클러스터링을 수행하였으며, 결과는 표 1에 있다.

표 1. splice 데이터셋에 대한 실험결과
Table 1. Clustering results for the splice dataset

클러스터 번호	알고리듬 1		알고리듬 2		제안하는 알고리듬	
	EI	IE	EI	IE	EI	IE
1	614	577	766	768	553	266
2	153	191	1	0	214	502

표 1에서 보면 알고리듬 1에서는 대부분의 시퀀스들이 클러스터 1에 몰려 있다. 또한, 클러스터 1에는 EI가 614개, IE가 577개, 클러스터 2에는 EI가 153개, IE가 191개로 EI와 IE가 대략 반반씩 섞여있다. 알고리듬 2로 클러스터링 한 결과는 1개의 시퀀스를 제외하고는 모든 시퀀스가 클러스터 1으로 클러스터링 되어 있다. 이에 반해, 제안하는 알고리듬에서는 클러스터 1에 EI가 553개, IE가 266개, 클러스터 2에 EI가 214개, IE가 502개로 구성이 된다. 즉, 대부분이 EI으로 구성된 하나의 클러스터와 IE로 구성된 또 하나의 클러스터를 얻을 수 있었다. 본 연구에서 제안하는 유사도를 사용함으로써 클러스터링 결과가 좋아진 것을 알 수 있었다.

5.2 합성 데이터셋

본 연구에서 제안하는 알고리듬의 성능을 평가하기 위해서, Quest 프로젝트의 합성 데이터 생성기[17]를 응용하여 표 2와 같은 4개의 합성 데이터셋을 생성하였으며, 이를 각각 DS1, DS2, DS3, DS4라 한다.

표 2. 합성 데이터셋
Table 2. Synthetic dataset
(단위: 트랜잭션 개수)

데이터셋의 종류 \ 클러스터 번호	1	2	3	4	5	아웃라이어	총 트랜잭션 수
DS1	175	160	235	340	800	10	1000
DS2	350	320	470	680	160	20	2000
DS3	525	480	705	1020	240	30	3000
DS4	700	640	940	1360	320	40	4000

실험은 표 2의 데이터셋을 5.1절에서처럼, 기존 알고리듬1, 2와 제안하는 방법으로 클러스터링을 수행하였다. 합성 데이터셋에서는 트랜잭션들이 어느 클러스터에 속하는지를 알고 있기 때문에, 오분류된 트랜잭션의 수를 쉽게 계산할 수 있으며, 따라서 이것을 클러스터링 결과의 평가 척도로 사용하였다. 그림 5는 알고리듬 1, 2와 제안하는 방법으로 클러스터링을 수행하였을 경우에 오분류된 트랜잭션의 수를 나타낸다.

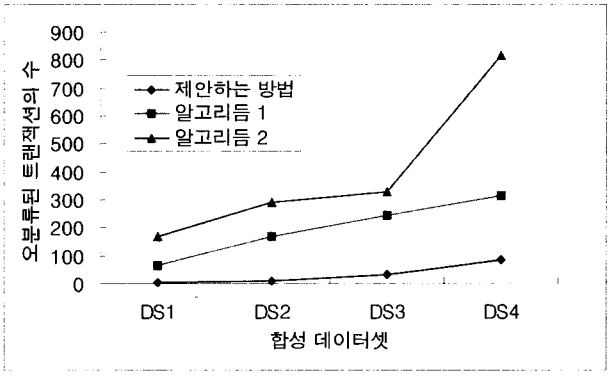

그림 5. 합성 데이터셋에 대한 실험 결과
Fig 5. Results for the synthetic dataset

또한, 클러스터의 개수를 변화시키면서 합성 데이터셋을 생성하여 알고리듬 1, 2와 제안하는 방법으로 클러스터링을 수행하였으며, 결과는 그림 6과 같다.

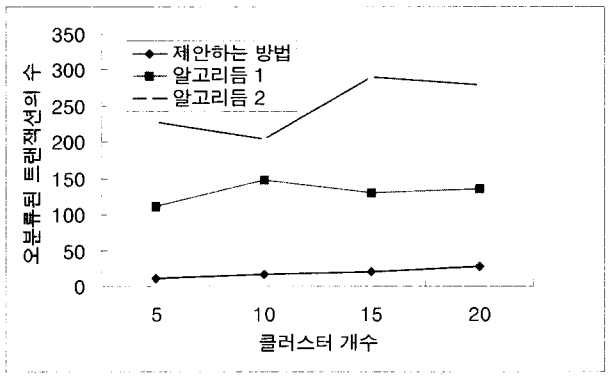

그림 6. 클러스터 개수 변화에 대한 실험 결과
Fig. 6. Clustering results with respect to the number of clusters

다음으로, 표 3과 같은 합성 데이터셋을 이용하여, 샘플수와 근접 이웃 수(k)를 변화시키면서 제안하는 클러스터링 방법으로 실험을 하였다. (그림 7 참조)

표 3. 합성 데이터셋
Table 3. Synthetic dataset

클러스터 번호	1	2	3	4	5	아웃라이어	합계
트랜잭션 개수	17500	16000	23500	34000	8000	1000	100000

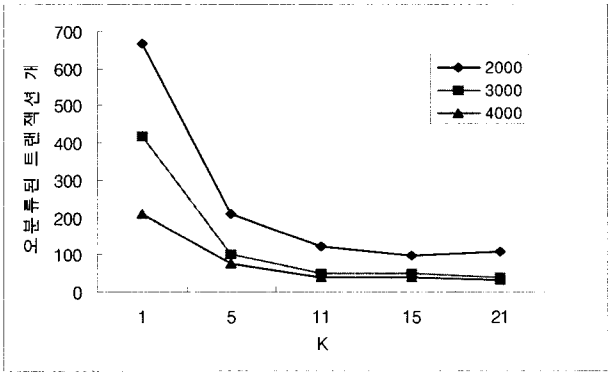

그림 7. 합성 데이터셋에 대한 실험 결과
Fig 7. Results for the synthetic dataset

그림 7에서 보면, 샘플의 크기가 커질수록 원래의 클러스터들을 정확히 찾아낸다. 또한, k 값이 11이상 되면, k 값이 1일 경우보다 오분류 트랜잭션의 개수가 현저히 줄어듬을 알 수 있다.

표 1과 그림 5, 6의 실험 결과에서 사용한 데이터셋들은 시퀀스들의 총 개수가 작은 관계로 3가지 알고리듬 모두 샘플링과 k-nn을 이용하지 않고, 모든 시퀀스들을 대상으로 클러스터링을 수행하였다. 그러므로, 제안하는 클러스터링 방법에서 기존 알고리듬들 보다 좋은 결과를 얻은 이유는 제안하는 유사도 계산방법이 기존 방법들 보다 우수하기 때문이다. 또한, 그림 7은 샘플링과 k-nn방법을 이용한 새로운 클러스터링 방법에 대한 실험 결과이다. 여기서는 기존 방법들처럼 단순히 유사도가 높은 하나의 데이터만을 이용하여 클러스터링 하는 것보다 k-nn을 이용(k>=11)함으로써 오분류 데이터의 수가 줄어듬을 알 수 있다.

6. 결 론

본 논문에서는 범주형 항목들의 순서를 고려한 시퀀스들의 클러스터링 문제를 연구하였다. 본 문제를 풀기 위하여 새로운 유사도 계산 방법을 제안하였다. 시퀀스들 간의 유사도는 순서를 가지는 두 항목 쌍들이 비교 대상의 두 시퀀스들 사이에 얼마나 많이 포함되어 있느냐에 따라 계산이 된다. 또한, 유사도를 효율적으로 계산하는 방법도 제안하였다.

대규모의 데이터들을 클러스터링 하는 경우에는 계층적 클러스터링 알고리듬은 높은 계산량 때문에 적용이 불가능하기에, 본 연구에서는 샘플링과 k-nn 방법을 이용한 새로운 클러스터링 방법을 제안하였다. 마지막으로, splice 데이터셋과 합성 데이터셋을 이용한 실험을 통하여, 제안하는 방법이 기존의 방법들 보다 성능이 우수함을 보였다.

향후에는 다양한 데이터셋들에 대해 본 연구에서 제안하

는 알고리듬을 적용하는 것이 필요하며, 범주형뿐만 아니라 수치형 값들을 포함하는 시퀀스들도 연구해야 할 과제이다.

참고문헌

[1] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, pp. 335-393, 2001.

[2] M. Perkowitz and O. Etzioni, "Towards Adaptive Web Sites: Conceptual Framework and Case Study", Proc. 8th Int. WWW Conf., Canada, 1999.

[3] D. Gusfield, Algorithm on Strings, Trees, and Sequences, Press Syndicate of the University of Cambridge, New York, 1997.

[4] D. S. Hirschberg, Pattern Matching Algorithms, Oxford University Press, pp. 123-142, 1997.

[5] P. Moen, Attribute, Event Sequence, and Event Type Similarity Notions for Data Mining, Ph.D. Thesis, University of Helsinki, Dept. of Computer Science, 2000.

[6] K. Charter, J. Schaeffer and D. Szafron, "Sequence alignment using FastLSA", Proc. 2000 Int. Conf. Math and Eng. Tech. in Medicine and Biological Sci., Nevada, pp. 239-245, 2000.

[7] S. Guha, R. Rastogi and K. Shim, "CURE: An Efficient Clustering Algorithm for Large Databases", Information Syst., Vol. 25, No. 1, pp. 35-58, 2001.

[8] J. Han, M. Kamber and A. 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.

[9] S. Guha, R. Rastogi and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes", Information Syst., Vol. 25, No. 5, pp. 345-366, 2000.

[10] K. Wang, C. Xu and B. Liu, "Clustering Transactions Using Large Items", ACM CIKM Int. Conf. Information and Knowledge Management, pp. 483-490, 1999.

[11] R. Agrawal and R. Srikant, "Mining Sequential Patterns", Proc. Int. Conf. Data Engineering, Taiwan, 1995.

[12] M. Joshi, G. Karypis and V. Kumar, "Universal Formulation of Sequential Patterns", Technical Report TR 99-021, University of Minnesota, 1999.

[13] T. Morzy, M. Wojciechowski and M. Zakrzewicz, "Scalable Hierarchical Clustering Method for Sequences of Categorical Values", Proc. 5th Pacific-Asia Conf. KDD, Hong Kong, 2001.

[14] B. Hay, G. Wets and K. Vanhoof, "Clustering Navigation Patterns on a Website Using a Sequence Alignment Method", 2001 Int. Joint Conf. on Artificial Intelligence, 2001.

[15] W. Wang and O. R. Zaiane, "Clustering Web Sessions by Sequence Alignment", 13th Int. Workshop on Database and Expert Syst. Applications, France, 2002.

[16] C. L. Blake and C. J. Merz, UCI repository of machine learning databases, 1998.

[17] R. Agrawal, M. Mehta, J. Shafer, R. Srikant, A. Arning and T. Bollinger, "The Quest Data Mining System", Proc. 2nd Int. Conf. KDD, Portland, 1996.

저 자 소 개

오승준 (Seung-Joon Oh)
현재 : 한양대학교 산업공학과 박사과정 재학중

관심분야 : 데이터 마이닝, 인공지능
Phone : 02-2290-0474
Fax : 02-2290-0471
E-mail : hiosj@ihanyang.ac.kr

김재련 (Jae-Yearn Kim)
현재 : 한양대학교 산업공학과 교수

관심분야 : 데이터 마이닝, 전문가 시스템, 시뮬레이션
Phone : 02-2290-0474
Fax : 02-2290-0471
E-mail : jyk@hanyang.ac.kr