시퀀스 요소 기반의 유사도를 이용한 시퀀스 데이터 클러스터링

Mining Clusters of Sequence Data using Sequence Element-based Similarity Measure

오승준, 김재련

한양대학교 산업공학과
서울시 성동구 행당동 17, 133-791
Tel: +82-2-2290-0474, E-mail: hiosj@ihanyang.ac.kr, jyk@hanyang.ac.kr

Abstract

Recently, there has been enormous growth in the amount of commercial and scientific data, such as protein sequences, retail transactions, and web-logs. Such datasets consist of sequence data that have an inherent sequential nature. However, only a few of the existing clustering algorithms consider sequentiality. This study presents a method for clustering such sequence datasets. The similarity between sequences must be decided before clustering the sequences. This study proposes a new similarity measure to compute the similarity between two sequences using a sequence element. Two clustering algorithms using the proposed similarity measure are proposed: a hierarchical clustering algorithm and a scalable clustering algorithm that uses sampling and a k-nearest neighbor method. Using a splice dataset and synthetic datasets, we show that the quality of clusters generated by our proposed clustering algorithms is better than that of clusters produced by traditional clustering algorithms.

Keywords: Clustering; Sequence; Similarity

1. 서론

클러스터링이란 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 [7]. 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 로그, 단백질 시퀀스, 소매점 거래 데이터 등과 같은 분야의 데이터들은 순서적인 면을 가지고 있는 시퀀스 데이터(또는 시퀀스)들이다. 즉, 데이터의 항목들 간에 순서가 존재하는 것이다. 예를 들어, 두 개의 시퀀스들이 동일한 항목들로 이루어졌더라도 항목들 간의 순서가 다르면 서로 다른 시퀀스들이다. 그러나, 기존의 클러스터링 방법들은 데이터들 내에 순서가 존재하는 면을 고려하지 않았거나,

효율적으로 시퀀스들 간의 유사도를 계산하는 방법을 사용하지 않았다.

항목들 간에 순서가 존재하는 시퀀스들을 클러스터링 하는 것은 많은 면에서 유용하다. 예를 들면, 웹 사용자들의 사이트 방문 기록을 보관한 웹 로그 파일들을 이용하여 웹 사용자들을 클러스터링 하는 것은 서로 다른 웹 사용자 그룹들을 발견하는데 도움을 준다[15]. 또한, 비슷한 구조를 공유하는 단백질 시퀀스들끼리 그룹화 하는 것은 비슷한 기능을 갖는 단백질 시퀀스들을 찾는데 도움을 준다.

본 연구에서는 웹 로그나 단백질 시퀀스, 소매점 거래 데이터 등과 같이 항목들 사이에 순서가 존재하는 시퀀스들을 클러스터링 하는 문제를 다룬다. 이를 위해서는 시퀀스들 간의 유사도를 구하는 것이 무엇보다 중요하다. 이를 위해, 본 연구에서는 기존의 유사도 계산 방법과 다른 새로운 유사도 계산 방법을 제안한다. 또한, 이 방법을 이용한 계층적 클러스터링 알고리듬도 제안한다.

대규모의 데이터들을 클러스터링 하는 경우에는 계층적 클러스터링 알고리듬은 계산량이 커서 적용이 불가능하므로 새로운 클러스터링 방법이 요구된다. 본 연구에서는 샘플링과 k-nearest neighbor(k-nn)방법을 이용하여 대규모의 데이터들에도 적용이 가능한 새로운 클러스터링 방법을 제안한다.

2. 기존 연구

다양한 클러스터링 기법들에 대한 연구들은 Han et al.[7]과 Ye[18]에 있으며, 클러스터링 기법들에 사용되는 데이터의 종류들에 대한 분류는 Han et al.[7]에 있다. 기존의 클러스터링 기법들은 주로 수치형 값들의 데이터[5][8]와 범주형 값들의 데이터[6]들만을 문제영역으로 다루어 왔다.

최근에는 웹 마이닝 분야에서도 클러스터링 기법을 이용한 연구가 활발히 이루어지고 있는데[14], 여기에는 비슷한 내용의 웹 페이지끼리 클러스터링을 하는 웹 contents 마이닝 분야의 Roussinov et al. [16]와 웹 사용자의 웹 사용 패턴을 클러스터링 하는 웹 usage 마이닝 분야의 Fu et al. [4]과 Mobasher et al. [13]이 있다. 그러나, Fu et al. [4], Mobasher et al. [13], Roussinov et al. [16] 모두 항목들 간의 순서는 고려하지 않고 있다.

시퀀스에 대한 연구는 주로 빈발하는 순차 패턴을 찾는데 집중되어 왔다. 이 문제는 Agrawal and Srikant [2]에서 처음으로 제안되었는데, 이 분야의 순차패턴을 탐사하는 문제는 시퀀스의 지지도가 사용자가 정의한 최소지지도보다 큰 시퀀스를 발견하는 것이다. Joshi et al. [11]에서는 순차 패턴을 일반화 시켜 표현하는 방법을 다루었다

시퀀스들에 대한 클러스터링 연구로는 다음의 세 가지 연구가 있다. Morzy et al. [14]은 빈발패턴이 주어져 있다고 가정을 하고, 이 빈발 패턴을 하나 이상 포함한 시퀀스들만을 대상으로 클러스터링을 수행한다. Hay et al. [9]은 시퀀스들 사이의 유사도로 edit distance 방법을 사용하여 클러스터링을 수행하고, Wang and Zaiane [17]는 sequence alignment 방법을 이용하여 클러스터링을 수행한다. 그러나, 본 연구에서는 Morzy et al. [14]와 달리 빈발패턴에 상관없이 모든 시퀀스들을 대상으로 클러스터링을 수행하고, Hay et al. [9], Wang and Zaiane [17]에서 사용한 유사도 계산 방법과 다른 새로운 유사도를 사용하여 시퀀스들을 클러스터링 한다.

계층적 클러스터링 방법의 단점을 보완하기 위하여 샘플링을 이용한 연구로는 Guha et. al. [5][6]가 있다. 두 알고리듬 모두 전체 데이터가 아니라 랜덤 샘플링을 이용하여 추출한 샘플들에 대해서만 계층적 알고리듬을 적용한다. 샘플이외의 데이터들은 계층적 알고리듬에 의하여 결정된 클러스터들중 가장 가까운 클러스터에 데이터들을 할당한다. 그러나, 이들은 시퀀스 데이터가 아니라 범주형이나 수치형 값들로만 이루어진 데이터들을 대상으로 클러스터링을 수행한다.

3. 시퀀스들 간의 유사도 계산 방법

3.1 유사도 측정

데이터들을 그룹화하거나 클러스터링 하는데 있어서는 유사도의 개념이 중요하다. 그러나, 데이터들 사이의 유사도는 데이터의 종류에 따라 달라지며 데이터의 특성에 따라 여러 종류의 유사도 측정 방법들이 존재한다. 즉, 두 데이터들 사이의 유사도가 어떤 유사도 측정 방법에서는 매우 높게 나올 수 있지만, 다른 유사도 측정 방법을 이용하면 낮게 나올 수도 있는 것이다.

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 서브 셋들이 얼마나 많이 존재하느냐를 고려한다. 본 연구에서는 동일 서브 셋들을 찾기 위해 순서를 가지는 두 항목 쌍들을 이용한다. 즉, 두 시퀀스들 사이에 동일 항목 쌍들이 많을수록 유사도가 높게 나오는 성질을 이용한다.

[예제 3.1] 두 시퀀스 S_1 = <A B C D>, S_2 = <A C D E> 가 있다. S_1 의 두 항목 쌍들의 모임은 (AB, AC, AD, BC, BD, CD)이고 S_2 의 두 항목 쌍들의 모임은 (AC, AD, AE, CD, CE, DE)이다. S_1, S_2 에 동일한 두 항목 쌍들이 많을수록 유사도는 높다. 여기서,(AC, AD, CD)가 공통 두 항목 쌍들이다. □

3.2 시퀀스 요소를 이용한 제안하는 유사도 계산 방법

시퀀스 $S = < x_1\ x_2 \cdots x_i\ x_j \cdots x_n >$ 에서 순서를 가지는 2 개의 항목들로 구성된 $x_i\ x_j$ 를 시퀀스 요소 e_k 라고 하며, e_k 들의 모임을 $E = (e_1, e_2, \ldots, e_k, \ldots)$ 라 한다. E 의 크기는 E 에 있는 요소들의 개수이며, | E | 로 나타낸다.

[예제 3.2] 시퀀스 S = < A B C E >에서 시퀀스 요소들의 모임은 E = (AB, AC, AE, BC, BE, CE)이며, | E | = 6 이다. □

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

[정의 3.1] 두 시퀀스 S_1 과 S_2 의 시퀀스 요소들의 모임을 각각 E_1,E_2 라고 하면, S_1, S_2 의 유사도 $sim(S_1, S_2)$는 다음과 같이 정의한다.

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

여기서, $|E_1 \cap E_2|$는 E_1 과 E_2 의 공통 요소들의 개수이며, E_1 과 E_2 사이에 공통 항목들이 많을수록 유사도는 높고, 이 값을 $\frac{|E_1|+|E_2|}{2}$로 나누는 것은 유사도를 0 과 1 사이의 값을 갖도록 하기 위해서이다. □

[예제 3.3] 두 시퀀스 S_1 = < A B D A >, S_2 = < A C D A C >에서 시퀀스 요소들의 모임은 각각 E_1 = (AB, AD, AA, BD, BA, DA) 과 E_2 = (AC, AD, AA, AC, CD, CA, CC, DA, DC, AC) 이며, | E_1 | = 6, | E_2 | = 10, $E_1 \cap E_2$ = (AD, AA, DA), | $E_1 \cap E_2$ | = 3 이다. 따라서, 두 시퀀스의 유사도 $sim(S_1, S_2)$는 3/8 이다. □

유사도 측정을 위하여 3 항목 이상의 시퀀스 요소를 사용 할 수도 있다. 그러나, 본 연구에서 2 항목 시퀀스 요소들만 고려하는 이유는 첫째, 3 항목 이상으로 구성된 시퀀스 요소들도 세분화해 보면 2 항목 시퀀스 요소들로 모두 표현할 수 있기 때문이다. 둘째, 2 항목 시퀀스 요소를 고려하는 것이 3 개 이상의 항목들로 시퀀스 요소들을 구성하는 것보다 계산량에 있어 훨씬 효율적이기

때문이다. 예를 들어, 시퀀스 S = < x_1 x_2 ⋯ x_i ⋯ x_n>의 2 항목 시퀀스 요소들의 수는 $_nC_2$ 이지만, 3 항목 시퀀스 요소들의 수는 $_nC_3$ 이 된다. n>5 에서는 2 항목 시퀀스 요소들의 개수보다 3 항목 시퀀스 요소들의 개수가 커지기 때문에 공통된 시퀀스 요소들을 찾는데 있어 계산량은 더욱 늘어나게 된다.

제안하는 방법에서는 시퀀스 요소를 이용하여 두 시퀀스들 사이의 유사도를 계산하므로, edit distance 방법처럼 다수의 edit operations 조합이 나오지 않는다. 또한, sequence alignment 방법처럼 scoring scheme 에 의존적이지 않으며, 항목 값들의 종류가 많은 경우에도 효율적으로 유사도를 측정할 수 있다.

4. 계층적 클러스터링 알고리듬

계층적 클러스터링 알고리듬은 통합(agglomerative) 방법과 분리(divisive) 방법으로 나눌 수 있다. 통합 방법은 처음에 각각의 객체들을 하나의 클러스터로 설정 한 후 이들 쌍간의 거리 (혹은 유사도)를 기반으로 가장 가까운 클러스터(객체)들끼리 합병을 수행한다. 최종적으로 한 클러스터 내에 모든 객체들이 포함될 때까지 위의 과정을 반복한다. 분리 방법은 통합 방법과 반대로 위의 과정을 진행한다[7].

본 연구에서는 통합 방법의 계층적 클러스터링 알고리듬을 사용한다. n 개의 시퀀스들을 클러스터링 하는 문제를 생각해 보자. 처음에는 n (n-1)/2 개의 클러스터간 합병을 고려할 수 있는데, 이중에서 합병을 했을 경우 가장 높은 평가함수 값을 주는 두 개의 클러스터를 합병한다. *l* 번째 합병 후에는 (n-l) (n-*l*-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(2)를 사용한다.

$$\text{Maximize Cf} = \sum_{r=1}^{k} \frac{1}{n_r} \sum_{i,j \in C_r} \text{sim}(i,j) \quad (2)$$

여기서, n_r은 C_r 내의 시퀀스들 개수,
k 는 클러스터 개수

식(2)는 Zho et al. [19]에 있는 평가함수들 중 하나인 식(3)을 변형한 것이다. 식(3)은 모든 클러스터에 대하여, 클러스터내에 있는 데이터 쌍들 간의 유사도 평균에 데이터 개수를 곱하여 모두 합한 값이며, 데이터들 간의 유사도로 코사인 유사도를 이용했지만, 본 연구에서는 3 장에서 제안한 유사도 계산 방법을 사용한다.

$$\text{Maximize Cf} = \sum_{r=1}^{k} n_r \left\{ \frac{1}{n_r^2} \sum_{i,j \in C_r} \cos(i,j) \right\} \quad (3)$$

여기서, n_r은 C_r 내의 데이터들 개수,
k 는 클러스터 개수

본 연구에서는 최단 거리법(shortest linkage method), 평균 거리법 (average linkage method), 최장 거리법(complete linkage method) 등 기존의 방법들 대신에 식(2)를 평가함수로 사용한다.

본 연구에서 제안하는 클러스터링 알고리듬의 단계는 그림 1 과 같다.

Step 0. Initialize

For all i, C_i S_i ($\in$ D)

Step 1. Compute the value of the Cf

For each C_i, $C_j \in$ D,

compute the value of the Cf (Eq. (2))

Step 2. Merge

$C_{new} \leftarrow$ merge (C_i, C_j) for the max value of the Cf

Step 3. Check the condition

If ($|C_i| >$ k) then go to Step 1.

Else go to Step 4.

Step 4. Exit

그림 1. 계층적 클러스터링 알고리듬

Step 0 은 초기화 단계로서 데이터베이스 D 를 액세스하여 각각의 시퀀스를 하나의 클러스터로

설정한다. Step 1 은 두 클러스터가 합병이 될 경우의 평가함수 식(2)의 값을 구하는 단계로, 현재 n 개의 클러스터가 있다고 하면, n (n-1)/2 개의 평가함수 값을 계산한다. Step 2 는 합병 단계로서, Step 1 에서 계산한 평가함수 값들 중 가장 큰 값을 주는 두 개의 클러스터를 합병한다. Step 3 은 조건 검사 단계로서 클러스터의 개수가 지정된 클러스터 개수보다 크면 Step 1 로 간다. 그렇지 않으면 Step 4 로 간다. 마지막으로, Step 4 는 종료 단계로서 알고리듬을 끝낸다.

5. 샘플링과 k-nearest neighbor(k-nn)을 이용한 클러스터링 알고리듬

5.1 제안하는 클러스터링 알고리듬의 개요

대규모의 데이터들을 클러스터링 하는 경우에는 계층적 클러스터링 알고리듬은 계산량이 커서 적용이 불가능하므로, 그림 2 와 같이 새로운 클러스터링 알고리듬을 제안한다.

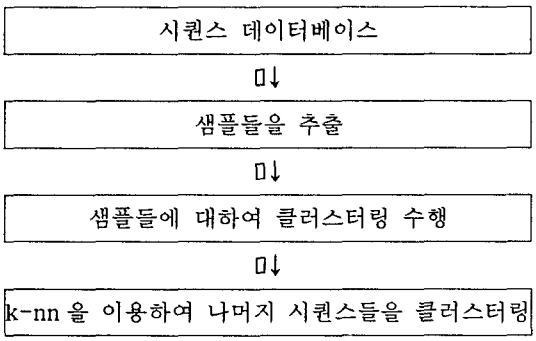

그림 2. 제안하는 클러스터링 방법의 개요

그림 2 에서 보면, 시퀀스 데이터베이스로부터 랜덤 샘플링을 이용하여 샘플들을 추출한다. 샘플들에 대해서는 계층적 클러스터링 알고리듬을 수행한다. 다음으로, 샘플들만으로 이루어진 클러스터들과 나머지 시퀀스들을 k-nearest neighbor (k-nn) 방법을 이용하여 클러스터링 한다.

5.2 랜덤 샘플링 단계

데이터 분석에 있어서 샘플링 과정은 다음과 같은 두 가지 방향에서 발생이 된다. 하나는 데이터 자체가 모집단으로부터의 단순한 샘플일 경우이며, 샘플링은 데이터를 수집하는 과정의 일부분이다. 이런 종류의 샘플링은 데이터 마이닝 분야에서는 관심 밖의 분야이다[7].

또 다른 하나는 최초의 데이터 셋이 매우 규모가 커서 샘플들을 가지고 데이터를 분석해야 할 경우이다. 이 경우에는 샘플들이 추출된 후, 이들이 전체 데이터에 대한 필요한 정보를 제공하게 된다.

랜덤 샘플링에서는 전체 데이터 셋에 있는 데이터들이 샘플들로 뽑힐 수 있는 가능성이 모두 동일하다. 여기에는 샘플로 뽑힌 데이터들이 다시 샘플로 뽑힐 수 있느냐 없느냐에 따라, 복원 랜덤 샘플링(random sample with replacement)과 비복원 랜덤 샘플링(random sampling without replacement) 등 두 가지 방법이 있다.

클러스터링 분야에서도 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[5][6]. 본 연구에서도 대규모의 데이터 셋을 효율적으로 처리하기 위해 랜덤 샘플링을 이용하여 샘플들을 추출한 후, 이들을 이용한다.

5.3 샘플들을 클러스터링 하는 단계

본 단계에서는 샘플들을 대상으로 클러스터링을 수행한다. 여기서는 4 장의 계층적 클러스터링 알고리듬을 사용한다.

5.4 k-nn 을 이용하여 나머지 시퀀스들을 클러스터링 하는 단계

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

분류 기법에는 의사결정 나무나 베이지안 분류기, k-nn 을 이용한 방법, 사례기반 추론, 러프 셋을 이용한 방법 등 여러 종류가 있다[10]. 이중에서 k-nearest neighbor 분류 기법은 패턴 분류 문제 영역에 있어서 잘 알려진 분류 기법중 하나이다. 특히, 중요한 특성중의 하나가 데이터들 간의 거리 또는 유사도만이 필요하다는 것이다.

본 단계의 알고리듬은 그림 3 과 같다.

Step 0. . Let S_i is the sequence to be clustered. Compute the similarity between S_i and each sequence in the clusters.

Step 1. Search for the k nearest neighbors that are closest to S_i

Step 2. Choose the cluster that contains the most k-nearest-neighbors.
If an equal number of k-nearest-neighbors exists, choose one cluster randomly.

Step 3. Assign S_i to the cluster selected in Step 2.

Step 4. Go to Step 0 if sequences remain to be clustered,
otherwise exit the algorithm

그림 3. 나머지 시퀀스들을 클러스터링 하는 알고리듬

Step 0 에서는 시퀀스 S_i 와 클러스터들에 속한 모든 시퀀스들 간의 유사도를 구한다. 이때, 시퀀스들 간의 유사도는 3 장에서 제안한 유사도 계산 방법을 사용한다. Step 1 에서는 시퀀스 S_i 에 대하여 유사도가 가장 높은 순서대로 k 개의 이웃 시퀀스들을 구한다. Step 2 에서는 Step 1 에서 구한 k 개의 이웃 시퀀스들이 가장 많이 속해 있는 클러스터를 구한다. Step 3 에서는 시퀀스 S_i 를 Step 2 에서 구한 클러스터에 할당한다. Step 4 에서는 클러스터링 하고자 하는 시퀀스가 남아 있는지를 검사하여, 시퀀스가 남아 있으면 Step 0 으로 가고, 그렇지 않으면 알고리듬을 끝낸다.

6. 실험결과

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

6.1 splice 데이터 셋

splice 데이터 셋은 UCI KDD 아카이브에 포함되어 있는 데이터 셋이다[3]. 이 데이터 셋은 60 개의 항목을 가진 뉴클레오타이드(nucleotide) 시퀀스들을 포함하고 있으며, 각각의 시퀀스들은 엑손/인트론 경계 (exon/intron, EI 라 부름)나 인트론/엑손 경계 (intron/exon, IE 라 부름)에 속하는 클래스 레이블을 가진다. EI 에 속하는 시퀀스들이 767 개이며, IE 에 속하는 시퀀스들이 768 개이다.

splice 데이터 셋을 3 가지 클러스터링 알고리듬으로 실험을 수행하였다. 알고리듬 1 은 시퀀스들 간의 유사도로 Hay *et al.* [9]처럼 edit distance 방법을 이용했으며, 본 연구에서 제안하는 계층적 클러스터링 알고리듬을 사용하여 클러스터링을 수행하였다. 알고리듬 2 는 시퀀스들 간의 유사도로 알고리듬 1 과 동일한 방법을 사용하였으며, 최장거리법을 이용한 계층적 클러스터링 방법을 사용하였다.

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

표 1. splice 데이터 셋에 대한 실험결과

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

6.2 합성 데이터 셋

본 연구에서 제안하는 알고리듬의 성능을 가하기 위해서, Quest 프로젝트의 합성 데이터 성기[1]를 응용하여 표 2 와 같은 합성 데이터 셋을 생성하였다.

표 2. 합성 데이터 셋

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

합성 데이터 셋은 트랜잭션으로 구성된 시장바구니 데이터베이스이며, 트랜잭션들의 항목 개수는 평균이 20 인 포아송 분포를 따른다. 또한, 총 트랜잭션의 1%가 아웃라이어들이며, 나머지 트랜잭션들은 5 개의 클러스터들 중 하나에 속한다.

합성 데이터 셋에서는 트랜잭션들이 어느 클러스터에 속하는지를 미리 알고 있기 때문에, 오분류된 트랜잭션의 수를 쉽게 계산할 수 있으며, 따라서 이것을 클러스터링 결과의 평가 척도로 사용하였다. 그림 4 는 샘플 수와 근접 이웃 수(k)의 변화에 따른 제안하는 클러스터링 방법의 실험결과를 나타낸다.

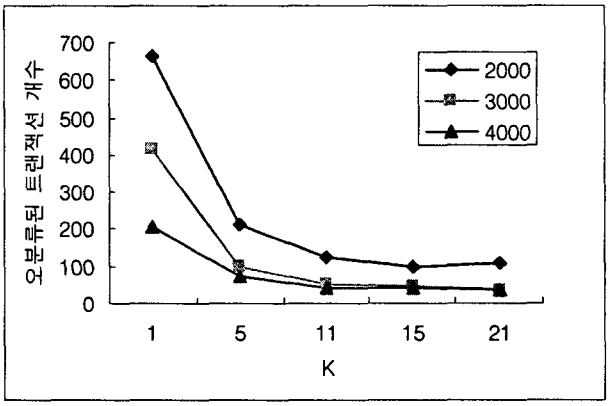

그림 4. 합성 데이터 셋에 대한 실험 결과

그림 4 에서 보면, 샘플의 크기가 커질수록 원래의 클러스터들을 정확히 찾아낸다. 또한, k 값이 11 이상 되면, k 값이 1 일 경우보다 오분류 트랜잭션의 개수가 현저히 줄어듬을 알 수 있다. 기존 방법들처럼 단순히 유사도가 높은 하나의 데이터만을 이용하여 클러스터링 하는 것보다 k-nn 을 이용(k>=11)함으로써 오분류 데이터의 수를 줄일 수 있었다.

7. 결론

본 논문에서는 범주형 항목들이 순서를 가지고 있는 시퀀스들의 클러스터링 문제를 연구하였다. 최근 들어 웹 로그, 단백질 시퀀스, 소매점 거래 데이터 등과 같은 데이터들의 폭발적인 증가를 볼 수 있다. 이런 데이터들은 항목들 간의 순서를 고려해야 하는 시퀀스 데이터들이다. 시퀀스들은 동일한 항목들로 이루어졌더라도 항목들 간의 순서가 다르면 서로 다른 시퀀스 데이터들이다. 그러나, 항목들 간의 순서적인 면을 고려한

클러스터링 연구는 많지 않았다. 본 논문에서는 이러한 시퀀스 데이터들을 클러스터링 하기 위한 문제를 연구하였다.

시퀀스 데이터들을 클러스터링 하기 위해서는 두 시퀀스들 사이의 유사도를 효율적으로 측정하기 위한 방법이 필요하다. 시퀀스들 간의 유사도를 측정하기 위해, 본 논문에서는 시퀀스 요소를 이용하는 새로운 유사도 측정 방법을 제안하였다.

본 논문에서는 두 시퀀스들 간의 유사도 계산 방법을 이용하여 시퀀스들을 클러스터링 하는 두 가지 방법을 제안하였다. 하나는 계층적 클러스터링 알고리듬이며, 다른 하나는 샘플링과 k-nearest neighbor (k-nn)을 이용한 클러스터링 알고리듬이다.

계층적 클러스터링 알고리듬은 처음에 각각의 시퀀스들을 하나의 클러스터로 설정 한 후, 각 클러스터들 간의 합병 시 주어진 평가함수의 값을 구한다. 이때 가장 높은 평가함수 값을 주는 두 클러스터들을 합병한다. 최종적으로는 주어진 개수의 클러스터가 남을 때까지 위의 과정을 반복한다.

대규모의 데이터들을 클러스터링 하는 경우에는 계층적 클러스터링 알고리듬은 계산량이 커서 적용이 불가능하므로, 본 연구에서는 샘플링과 k-nn 방법을 이용한 새로운 클러스터링 방법을 제안하였다. 랜덤 샘플링을 이용하여 샘플들을 추출한 후 이들을 이용하였으며, k-nn 을 사용함으로써 단순히 유사도가 가장 높은 하나의 데이터만을 이용하는 기존 방법보다 정확하게 클러스터링을 수행 하였다. 마지막으로, splice 데이터 셋과 합성 데이터셋을 이용한 실험을 통하여, 제안하는 클러스터링 방법이 기존 방법보다 성능이 우수함을 보였다.

향후에는 다양한 데이터 셋들에 대해 본 연구에서 제안하는 알고리듬을 적용하는 것이 필요하며, 범주형뿐만 아니라 수치형 값들을 포함하는 시퀀스들도 연구해야 할 과제이다.

참고문헌

[1] Agrawal, R., Mehta, M., Shafer, J., Srikant, R., Arning A., and Bollinger, T. (1996). "The Quest Data Mining System", *Proc. 2nd Int. Conf. Knowledge Discovery and Data mining*, Portland, OR.

[2] Agrawal, R., and Srikant, R. (1995). "Mining Sequential Patterns", *Proc. Int. Conf. Data Engineering*, Taiwan.

[3] Blake, C. L. and Merz, C. J. (1998). UCI Repository of Machine Learning Databases.

[4] Fu, Y., Sandhu, K. and Shih, M. Y. (1999). "Clustering of Web Users based on Access Patterns", *Proc. 1999 KDD workshop on Web Mining*, San Diego, CA.

[5] Guha, S., Rastogi, R. and Shim, K. (2001). "CURE: An Efficient Clustering Algorithm for Large Databases", *Information Syst.* 26(1):35-58.

[6] Guha, S., Rastogi, R. and Shim, K. (2000). "ROCK: A Robust Clustering Algorithm for Categorical Attributes", *Information Syst.* 25(5):345-366.

[7] Han, J. and Kamber, M. (2001). *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 335-393.

[8] Han, J., Kamber M. and Tung, A. K. H. (2001). "Spatial Clustering Methods in Data Mining: A Survey", *Geographic Data Mining and Knowledge Discovery*, eds. H. J. Miller and J. Han (Taylor and Francis, New York).

[9] Hay, B., Wets, G. and Vanhoof, K. (2003). "Segmentation of Visiting Patterns on Web Sites using a Sequence Alignment Method", *Journal of Retailing and Consumer Services*, 10, 146-153.

[10] Hirschberg, D. S. (1997). *Pattern Matching Algorithms,* Oxford University Press, 123-142.

[11] Joshi, M., Karypis, G. and Kumar, V. (1999). "Universal Formulation of Sequential Patterns", Technical Report TR 99-021, Univ. of Minnesota, Dept. of Com. Sci.

[12] Kosals, R. and Blockeel, H. (2000). "Web Mining Research: A Survey", *ACM SIGKDD*, 2(1):1-15.

[13] Mobasher, B., Dai, H., Luo, T., Nakagawa, M., Sun, Y. and Wiltshire, J. (2002). "Discovery of Aggregate Usage Profiles for Web Personalization", *Data Mining and Knowledge Discovery*, 6, 61-82.

[14] Morzy, T., Wojciechowski, M. and Zakrzewicz, M. (2001). "Scalable Hierarchical Clustering Method for Sequences of Categorical Values", *Proc. 5th Pacific-Asia Conf. Knowledge Discovery and Data Mining,* Kowloon, Hong Kong.

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

[16] Roussinov, D. and Zhao, J. L. (2003). "Automatic discovery of similarity relationships through web mining", *Decision Support Syst.* 35(1).

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

[18] Ye, N. (2003). *The handbook of data mining*, Lawerence Erlbaum Associates, New Jersey.

[19] Zho, Y. and Karypis, G. (2002). "Comparison of Agglomerative and Partitional Document Clustering Algorithms", *2nd SIAM Int. Conf. Data Mining*, Arlington, VA.