대용량 순차 데이터베이스에서 근사 순차패턴 탐색

금 혜 정[†] · 장 중 혁[††]

요 약

순차패턴 탐색은 다양한 응용 분야에서 매우 중요한 데이터 마이닝 작업으로 간주된다. 그러나 기존의 순차패턴 탐색 방법들은 길이가 긴 순차패턴이나 노이즈 정보를 다수 포함한 데이터베이스에 대한 마이닝에서는 한계가 있다. 해당 방법들은 매우 짧고 사소한 패턴들은 탐색하지만 다수의 순차 정보들에서 공유되는 중요 패턴들을 분석하는데 어려움을 겪는다. 본 논문에서는 이러한 문제를 해결하기 위한 방법으로 대용량 데이터베이스에 대한 근사 순차패턴 탐색 방법을 제안한다. 근사 순차패턴은 다수의 순차 정보들에서 근사적으로 공유되는 순차패턴을 의미한다. 제안된 방법은 두 과정으로 구분된다. 하나는 유사도에 따라 분석 대상 순차 정보들을 몇 개의 군집으로 나누는 과정이며, 다른 하나는 다중 정렬 방식을 적용하여 각 군집으로부터 대표 패턴을 찾는 과정이다. 이를 위해서 다수의 순차 정보들을 하나로 표현할 수 있는 가중치 순차패턴을 제시하며, 다수의 순차 정보들은 가중치 순차패턴 형태로 통합된다. 이렇게 통합된 정보를 가진 각 가중치 순차패턴을 이용하여 여러 순차 정보와 근사한 하나의 대표 패턴을 생성한다. 끝으로, 다양한 실험을 통해서 제안된 방법의 유용성을 검증한다.

키워드 : 근사 순차패턴, 순차패턴, 순차정보 군집화, 대표 패턴, 다중 정렬

Mining Approximate Sequential Patterns in a Large Sequence Database

Hye-Chung Kum[†] · Joong Hyuk Chang[††]

ABSTRACT

Sequential pattern mining is an important data mining task with broad applications. However, conventional methods may meet inherent difficulties in mining databases with long sequences and noise. They may generate a huge number of short and trivial patterns but fail to find interesting patterns shared by many sequences. In this paper, to overcome these problems, we propose the theme of *approximate sequential pattern mining* roughly defined as identifying patterns approximately shared by many sequences. The proposed method works in two steps: one is to cluster target sequences by their similarities and the other is to find consensus patterns that are similar to the sequences in each cluster directly through multiple alignment. For this purpose, a novel structure called weighted sequence is presented to compress the alignment result, and the longest consensus pattern that represents each cluster is generated from its weighted sequence. Finally, the effectiveness of the proposed method is verified by a set of experiments.

Key Words : Approximate Sequential Pattern, Sequential Patterns, Clustering Sequences, Consensus Pattern, Multiple Alignment

1. 서 론

일반적으로 순차패턴 탐색(Sequential pattern mining)이란 분석 대상 순차 데이터베이스에서 빈번히 발생되는 순차패턴들을 찾는 작업이다. 순차패턴의 개념은 [1]에서 제시된 이후 경제/기업 정보 분석, 웹 정보 마이닝 및 생물정보학 등의 분야에서 널리 활용되고 있다. 순차패턴 탐색은 데이터 마이닝 분야에서 매우 중요한 문제로 다뤄지고 있으며, 기존의 순차패턴 탐색은 다음과 같이 정의될 수 있다. 다수의 단위항목들로 구성되는 항목들이 순서적인 의미를 갖는 리스트를 구성하고 이들 리스트의 집합으로 정의되는 분석 대상 데이터베이스 및 기준 임계값인 최소 지지도가 주어졌을 때, 빈발 순차패턴 탐색은 해당 최소 지지도보다 큰 지지도를 갖는 부분 순차패턴을 탐색하는 작업이다[1, 2].

이전의 연구들에서 순차패턴 탐색에 대해서 널리 연구되고 다양한 방법들이 활발히 제안되었다. 하지만, 이들 방법들은 크게 두 가지 한계점을 지닌다. 첫째, 해당 방법들은 순차패턴 탐색시 정확한 일치에 기반하여 빈발 패턴을 탐색한다. 즉 하나의 순차패턴에 대해서 해당 순차패턴과 정확히 일치되는 부분 패턴을 갖는 순차 정보가 발생한 경우에만 해당 순차패턴의 출현빈도 수가 증가된다. 하지만 정확한 일치에 기반한 탐색 방법은 일반적으로 많은 노이즈(noise) 정보를 포함한 실제 응용 분야 데이터에서 충분히 긴 순차패턴을 탐색하지 못한다. 예를 들어 상품 구매 정보에 대한

† 정 회 원 : University of North Carolina, Chapel Hill 연구교수
†† 정 회 원 : 연세대학교 소프트웨어응용연구소 전문연구원
논문접수 : 2005년 10월 18일, 심사완료 : 2006년 1월 24일

분석에 있어서 다수의 고객들이 서로 유사한 구매 경향을 가졌다 할지라도 정확히 일치되는 경우는 매우 적다. 이러한 경우 의미적으로 중요한 충분히 긴 순차패턴은 거의 존재하지 않는다. 그런데 실제 응용 분야에서는 이와 같은 충분히 긴 순차패턴들이 평범한 짧은 순차패턴들 보다 훨씬 더 중요한 정보를 제공할 수 있다. 둘째, 기존의 순차패턴 탐색 방법들은 일반적으로 분석 대상 순차 데이터베이스에 존재하는 모든 순차패턴들을 구한다. 따라서 하나의 긴 순차패턴이 존재하는 경우 이보다 짧은 길이의 순차패턴들이 무수히 탐색된다. 예를 들어 길이가 20인 순차패턴에 대해서는 2^{20}-1개의 부분 순차패턴이 존재한다! 일반적으로 데이터 마이닝 작업은 데이터베이스를 분석하여 의미적으로 중요한 핵심 정보들을 얻는 것을 목적으로 한다. 이러한 목적을 고려할 때 마이닝 수행 결과 매우 많은 수의 순차패턴들을 제시하는 기존의 순차패턴 탐색 방법들은 유용성이 낮다고 할 수 있다. 이러한 단점을 보완하기 위해서 근래에는 closed frequent pattern[3] 등과 같이 보다 함축된 결과를 구해주는 마이닝 방법들이 빈발 패턴 탐색 분야에서 활발히 제안되었다. 하지만, 이들 방법들도 희소 데이터 집합(sparse data set)이나 노이즈 정보를 포함한 데이터 집합에 대해서는 유용한 분석 결과를 제공하지 못한다. 또한, 노이즈 정보가 있는 데이타 속에 정확한 일치를 바탕으로 한 함축된 결과는 유사한 결과를 통합하지 못하기 때문에 여전히 많은 수의 순차 패턴을 제시한다. 이러한 상황을 고려할 때 의미적으로 중요하면서도 충분히 긴 순차패턴을 제시하고 보다 축약된 형태의 정보를 제공할 수 있는 새로운 접근 방법을 필요로 한다.

본 논문에서는 대용량의 순차 데이터베이스에 대한 순차패턴 탐색에 있어서 효율적이며, 의미적으로 중요하면서도 충분히 긴 순차패턴을 구하는 새로운 순차패턴 탐색 방법을 제안한다. 이는 다음과 같은 특성을 갖는다.

- 논문에서 제안되는 방법은 정확한 일치에 기반한 기존의 순차패턴 탐색 방법들과는 달리 보다 유연한 접근 방법인 근사 일치에 기반하여 순차패턴을 탐색한다. 이를 통해서 실제 응용 분야에서 평범한 짧은 순차패턴들보다 유용하게 활용될 수 있는 의미적으로 중요하면서 충분히 긴 길이를 갖는 근사 순차패턴을 탐색한다.
- 논문에서 제안되는 방법은 마이닝 수행 결과 방대한 수의 순차패턴을 구하는 것이 아니라 분석 대상 순차 데이터베이스를 대표할 수 있는 소수의 대표 순차패턴(consensus sequential pattern)들을 구한다. 대표 순차패턴은 다수의 순차 정보들에서 공유되며, 짧은 순차패턴들을 포괄할 수 있는 것으로서 실제 응용 분야에서 보다 유용하게 활용될 수 있을 것이다.
- 논문에서 제안되는 방법에서는 두 개의 순차패턴간의 거리가 일정 임계값 이하인 경우 이들을 유사 순차패턴으로 간주하여 하나의 군집으로 간주하고, 해당 군집으로부터 하나의 대표 순차패턴을 구한다. 두 순차패턴간의 거리는 서로간의 유사도를 정의하기 위한 것으로서 상세한 의미는 3절에서 기술한다.
- 하나의 순차 정보 군집에 대해서 대표 패턴을 구하기 위해서 본 논문에서는 다수의 순차 정보를 효율적으로 통합 할 수 있는 다중 정렬 기법을 적용한다. 이렇게 다중 정렬된 (multiple alignment) 다수의 순차 정보는 하나의 가중치 순차패턴(weighted sequence)으로 통합되며, 각 가중치 순차패턴 중 주요 부분만 모아 대표 패턴을 생성한다.

본 논문의 구성은 다음과 같다. 먼저, 제 2장에서는 근사 순차패턴 탐색과 연관된 이전의 관련 연구들을 기술한다. 제 3장에서는 순차 정보 군집화 과정 및 개별 군집으로부터 대표 순차패턴을 구하는 과정 등으로 구성되는 근사 순차패턴 탐색 방법을 상세히 기술하고, 제 4장에서는 다양한 실험을 통해서 해당 방법의 유용성을 검증한다. 끝으로, 제 5장에서 논문의 결론을 맺는다.

2. 관련 연구

순차패턴 마이닝 방법들은 데이터 마이닝 분야에서 활발히 연구되고 있는 주요 관심 분야 중의 하나로서 다수의 탐색 방법들이 제안되어 왔다. 순차패턴 탐색을 위한 이전의 방법들 중 다수는 [1]에서 제안된 *Apriori* 속성에 기반을 두고 있다. 전형적인 *Apriori*-기반 방법은 다중 탐색 방법이다. 따라서, 해당 방법은 분석 결과 얻어지는 빈발 순차패턴의 최대 길이가 n일 때 분석 대상 데이터 집합을 $n+1$번 탐색 해야한다. 이들 방법들은 일반적으로 다음과 같은 특성을 갖는다[2]. 첫째, 마이닝 수행 과정에서 생성되는 후보항목 집합이 매우 크다. 둘째, 이들 방법들은 마이닝 결과를 얻기 위해서 분석 대상 데이터 집합을 다중 탐색해야 한다. 셋째, 방대한 수의 순차패턴 정보를 구하며, 중요성이 낮은 짧은 길이의 순차패턴이 다수를 이룬다. 한편, PrefixSpan[2]이나 SPAM[4] 등과 같이 깊이 우선 탐색에 의한 순차패턴 탐색 방법들도 다수 제안되었으며, 점진적으로 확장되는 순차 데이터베이스에서 효율적으로 순차패턴을 탐색하기 위한 방법도 [5]에서 제안되었다. 하지만 이들 방법들에서도 마이닝 결과 집합으로 중요성이 낮은 짧은 길이의 순차패턴들을 포함하는 방대한 수의 순차패턴들을 구한다. 즉, 앞서 기술한 바와 같이 대부분의 이들 방법들은 정확한 일치에 기반하여 순차패턴을 탐색하고 방대한 수의 마이닝 결과를 구한다. 이러한 특성은 이들 방법들에서 얻어진 순차패턴들이 실제 응용 분야에서 유용하게 활용되는데 장애가 될 수 있다.

원천 데이터를 분석하여 유용한 정보를 탐색하기 위한 새로운 접근 방법으로서 생물 정보학 분야에서 다중 정렬(multiple alignment) 기법이 활용되었으며[6-8], 해당 방법은 하나의 분석 대상 집합에서 공통된 패턴을 찾는 것을 목적으로 한다. 한편, 근사 빈발 패턴 탐색 방법은 [9] 및 [10]에서 연구되었다. [9]는 순서적인 정보를 고려하지 않는 빈발 항목집합(frequent itemset) 탐색 과정에서 근사 접근법을 적용한 연구이며 [10]은 문자열(string) 중에서 빈발 문자

열을 탐색하는 과정에서 근사 접근법을 적용한 연구로서, 이들 연구에서 제안된 방법들은 순차적인 정보를 고려하지 않거나 혹은 단위문자의 순서를 고려하는 정도에 그치고 있다. 하지만, 본 논문에서는 항목집합의 순서 정보를 고려하는 순차 데이터 집합에서 빈발 항목집합 순차 패턴(frequent sequence of itemsets)을 탐색하는 문제에 관심을 두고 있다. 일반적으로 항목집합의 순차 데이터 집합에서 빈발 패턴을 탐색하는 문제는 단순한 빈발 항목집합 탐색 문제보다 훨씬 더 복잡한 문제로서 빈발 항목집합 탐색 방법을 항목집합의 순서 정보를 고려하는 순차 데이터 집합에서 빈발 항목집합 순차 패턴을 탐색하는데 단순 적용하는 것은 불가능하며, 이들 방법들간의 직접적인 비교도 불가능하다. 따라서, [9] 및 [10]번에서 제안된 방법들은 본 논문에서 다뤄지는 항목집합의 순서를 고려하는 순차 데이터 집합에서 근사 빈발 항목집합 순차 패턴을 탐색하는 데 적용하는 것은 불가능하며, 이에 대한 상세한 설명은 생략한다.

3. 근사 순차패턴 탐색

본 절에서는 정확한 일치에 기반하여 순차패턴들을 구함으로써 실제 응용 분야에서의 활용성이 낮은 마이닝 결과를 구하는 기존의 순차패턴 탐색 방법들의 단점을 보완하기 위한 방법으로서 대용량의 순차 데이터베이스를 분석하여 의미적으로 중요하면서도 충분히 긴 순차패턴을 탐색하는 근사 순차패턴 탐색에 대해 상세히 기술한다. 해당 방법은 크게 두 과정으로 구분된다. 첫째, 분석 대상 순차 정보들을 각 순차 정보들간의 유사도에 기반하여 몇 개의 군집(cluster)으로 분류하는 순차 정보 군집화(clustering) 과정으로서, 3.1절에서 상세히 기술한다. 둘째, 각 순차 정보 군집으로부터 해당 군집을 나타내는 대표 패턴(consensus pattern)을 구하는 과정으로서, 3.2절에서 상세히 기술한다.

3.1 순차 정보 군집화

본 절에서는 분석 대상 순차 데이터베이스를 구성하는 순차 정보들을 상호간의 유사도를 근거로 다수의 군집으로 분류하는 순차 정보 군집화 과정에 대해서 기술한다.

순차 정보들간의 유사도. 일반적으로 순차 정보들간의 유사도를 구하기 위한 척도로써 *hierarchical edit distance*를 사용한다. 이는 하나의 순차 정보가 다른 순차 정보로 변화되는데 필요한 최소 전환 비용을 의미하며, 전환 비용이란 insertion, deletion 및 replacement 작업의 횟수를 의미한다. 이때, insertion 및 deletion은 기준 순차 정보 및 목적 순차 정보가 바뀔 뿐 서로 동일한 비용으로 간주할 수 있으며, 이를 통합하여 하나의 연산자 *INDEL()* 로 나타낸다. 한편 replacement 작업은 별도의 연산자 *REPL()*로 나타내면, 하나의 순차 정보에 포함되는 X를 Y로 변환하는 작업(즉, *REPL(X,Y)*)은 다음의 관계를 만족한다.

$$REPL(X, Y) \quad INDEL(X) + INDEL(Y)$$

두개의 순차 정보 $S_1=\langle X_1, \cdots, X_n\rangle$ 및 $S_2=\langle Y_1, \cdots, Y_m\rangle$가 주어졌을 때, 해당 순차 정보들간의 *hierarchical edit distance*는 다음과 같은 회귀 연산 관계에 의해 구할 수 있다.

$$D(0, 0) = 0$$
$$D(i, 0) = D(i-1, 0) + INDEL(X_i) \qquad (1 \le i \le n)$$
$$D(0, j) = D(0, j-1) + INDEL(Y_j) \qquad (1 \le j \le m)$$
$$D(i, j) = \min\begin{cases} D(i-1, j) + INDEL(X_i) \\ D(i, j-1) + INDEL(Y_j) \\ D(i-1, j-1) + REPL(X_i, Y_j) \end{cases}$$
$$(1 \le i \le n) \text{ and } (1 \le j \le m) \qquad (1)$$

이때, 서로 다른 길이의 순차 정보들에 대한 hierarchical edit distance를 서로 비교하기 위해서 각 순차 정보에서 구해진 hierarchical edit distance를 해당 순차 정보의 길이를 고려하여 정규화한다. 이를 normalized edit distance라 지칭하며, 다음과 같이 구한다.

$$dist(S_1, S_2) = \frac{D(n, m)}{\max\{|S_1|, |S_2|\}} \qquad (2)$$

식 (1)의 hierarchical edit distance는 순차 정보들간의 변환 과정에서 각 요소의 변환 비용에 근거하여 유사도를 구한다. 보다 일반적인 항목집합(itemset)들의 순차 정보들간의 유사도를 효율적으로 구하기 위해서는 변환되는 대상이 항목집합임을 고려하여 항목집합 간의 변환값을 측정해야 한다. 즉, 식 (1) 및 (2)를 이용하여 두 순차 정보들간의 hierarchical edit distance를 구하는데 있어서, 변환 대상이 항목집합으로 구성되는 순차 정보인 경우, 항목집합 X를 항목집합 Y로 변화하는데 필요한 비용 *REPL(X,Y)*는 다음과 같이 정의된다.

$$REPL(X, Y) = \frac{|(X-Y)\cup(Y-X)|}{|X|+|Y|} = \frac{|X|+|Y|-2|X\cap Y|}{|X|+|Y|}$$
$$(0 \le REPL(X,Y) \le 1) \qquad (3)$$

또한, *INDEL(X)*는 식 (3)에 의해서 다음과 같이 구해진다.

$$INDEL(X) = REPL(X,0) = 1$$

순차 정보 군집화. 식 (2)에서 구해진 hierarchical edit distance를 이용하여 분석 대상 순차 데이터베이스에 포함되는 순차 정보들을 유사도에 따라 몇 개의 군집으로 분류하며, 이때 밀도 기반 군집화(density-based clustering) 방법을 적용한다. 분석 대상 순차 데이터베이스에 포함되는 하나의 순차 정보에서 해당 순차 정보와 유사한 순차 정보들이 많을수록 해당 순차 정보는 밀집(dense) 순차 정보로 간주한다. 특히, 순차 데이터베이스 S의 각 순차 정보 S_i에 대해서 $dist(S_i, \neq S_j)$ ($S_i \neq S_j$, S_i and $S_j \in S$) 값 중에서 0이 아닌 가장 작은 k개의 값들을 $d_1, \cdots, d_k$로 나타낼 때, 해당 순차 정보의 *밀집도* *density*(S_i)는 다음과 같이 정의된다.

$$density(S_i) = \frac{n}{|S| \times d} \qquad (4)$$

여기서, $d=\max\{d_1, \cdots, d_k\}$이며 $n=|\{ S_j \in S \mid dist(S_i, S_j) \le d\}|$를 의미한다. 또한, n은 k-근린 공간(k-nearest neighbor space)에 포함되는 순차 정보의 수를 나타내며, k 값은 사용자에 의해 설정된다. 식 4에서와 같은 *밀집*도 함수를 바탕으로 [11]에서 제안된 알고리즘을 순차 정보 군집화에 적합하도록 개선하여 알고리즘 1을 얻는다.

알고리즘 1. (Uniform-kernel k-NN 군집화)

입력 : 순차 정보들로 구성되는 순차 데이터베이스 $S=\{S_i\}$, k : 근린(nearest neighbor) 순차 정보의 수

출력 : 군집들의 집합 $\{C_j\}$, 개별 군집은 다수의 순차 정보들로 구성

수행방법 :

1. (**초기화**) 각 순차 정보를 개별 군집으로 간주한다. 순차 정보 S_i로 정의되는 군집 C_{S_i}에 대해서 순차 정보 S_i의 *밀집*도를 구하여 해당 군집의 *밀집*도로 정의한다. 즉, $density(C_{S_i})=density(S_i)$.
2. (**순차 정보 밀집도 기반 통합**) 각 순차 정보 S_i에 대해서 $S_{i1}, \cdots, S_{in}$을 해당 순차 정보의 근린 순차 정보(nearest neighbor sequence)라 하자. n은 식 4에서 구해지는 값이다. 이때, 해당 순차 정보 S_i에 대해서, 자신의 밀집도 보다 큰 밀집도를 갖는 근린 순차 정보 S_j $(1 \le j \le n)$ 중에서 가장 가까이 있는 근린 순차 정보와 통합한다. 즉, $density(S_i)<density(S_j)$를 만족하는 근린 순차 정보 중에서 가장 가까이 있는 근린 순차 정보를 S_j라 한다면, C_{S_i}와 C_{S_j}를 통합한다. 통합된 군집의 *밀집*도는 $\max\{density(C_{S_i}), density(C_{S_j})\}$로 설정된다.
3. (**군집 밀집도 기반 통합**) 이어서 순차 정보 통합 과정을 통해 형성된 군집을 기준으로 하는 통합 과정을 수행한다. 하나의 순차 정보 S_i와 이를 포함하는 군집 C_{S_i}에 대해서, 해당 순차 정보의 밀집도보다 큰 밀집도를 갖는 근린 순차 정보들은 존재하지 않으나 해당 군집 C_{S_i}의 이웃 군집에 속하는 하나의 순차 정보 S_j의 밀집도가 순차 정보 S_i의 밀집도와 동일하고 순차 정보 S_j를 포함하는 군집 C_{S_j}의 밀집도가 군집 C_{S_i}의 밀집도보다 큰 경우 (즉, $density(C_{S_j})>density(C_{S_i})$ 및 $density(S_i)=density(S_j)$ 관계를 만족하는 경우), 두 군집 C_{S_j}와 C_{S_i}를 통합한다. 이는 고립된 군집들을 통합하기 위한 과정이다.

대용량의 순차 데이터베이스에 대한 군집화를 위한 알고리즘 1의 수행 결과에 가장 크게 영향을 미치는 중요한 매개변수는 근린 순차 정보(nearest-neighbor sequence)의 수를 정의하는 k 값이다. k 값이 큰 경우에는 보다 많은 수의 순차 정보들이 하나의 군집을 형성하며, 반면, k 값이 작은 경우는 보다 작은 수의 순차 정보들이 하나의 군집을 형성한다. k 값 조절을 통해서 보다 많은 순차 정보들을 포함하는 강한 군집을 생성하거나 또는 보다 작은 수의 순차 정보들을 포함하는 약한 군집을 생성할 수 있다.

3.2 다중 정렬에 의한 대표 패턴 탐색

순차 정보에 대한 군집화 과정에서 서로 유사한 순차 정보들은 동일한 군집으로 분류된다. 다음은 각 개별 군집에 속하는 순차 정보들을 효율적으로 통합하여 각 군집에 속하는 순차 정보들의 경향을 대표할 수 있는 대표 순차패턴을 탐색한다. 이를 위해서 본 절에서는 다중 정렬(multiple alignment) 방법을 이용하여 다수의 유사 순차 정보들을 하나의 대표 순차패턴으로 통합하는 과정에 대해서 기술한다.

3.2.1 순차 정보들의 다중 정렬

순차 정보들의 다중 정렬 과정은 하나의 군집에 속하는 다수의 순차 정보들을 하나로 통합하는 과정이다. 각 순차 정보들을 구성하는 단위 정보들의 종류와 위치를 고려하여 하나의 정보로 통합한다. 예를 들어 하나의 군집이 <표 1>에서와 같이 5개의 순차 정보로 구성되는 경우 각 순차 정보들을 구성하는 단위 정보(즉, 항목 집합)의 종류 및 위치를 고려하여 하나의 가중치 순차 정보로 통합한다. 이때, 각 단위 정보의 발생 빈도 및 하나의 군집에 속하는 순차 정보의 수 등도 함께 분석된다. 상세한 수행 과정은 다음과 같다.

먼저, 순차 정보들의 다중 정렬은 순차 정보의 군집화 과정에서 구해진 각 순차 정보들의 *밀집*도를 기준으로 수행된다. 즉, 하나의 군집에 속하는 순차 정보에 대한 다중 정렬 과정에서는 해당 순차 정보들을 *밀집*도 값을 기준으로 내림차순으로 정렬하고, 해당 순서에 따라 다중 정렬 과정을 수행한다.

한편, 순차 정보들의 다중 정렬 결과를 효율적으로 관리하기 위해서 본 논문에서는 다음과 같은 형태로 표현되는 가중치 순차패턴을 제안한다.

$$\text{가중치 순차패턴 } WS = \langle X_1{:}v_1, \cdots, X_l{:}v_l\rangle{:}n$$

여기서 각 기호들의 의미는 다음과 같다.

1. n은 해당 가중치 순차패턴의 전역 가중치를 나타내며, 해당 가중치 순차패턴이 n개의 순차 정보에 대한 정렬 결과 생성되었음을 의미한다.
2. $i(1 \le i \le l)$ 번째 항목집합으로 정렬되는 항목집합 X_i는 현재까지 정렬된 n개의 순차 정보들 중에서 v_i개의 순차 정보에 출현하였음을 의미한다.
3. WS를 형성하는 각 항목집합은 $X_i=(x_{j1}{:}w_{j1}, \cdots, x_{jm}{:}w_{jm})$ 형태를 띄고 있으며, 현재 정렬된 n개의 순차 정보들 중에서 $i(1 \le i \le l)$ 번째 위치에 정렬되는 항목집합에 $x_{jk}(1 \le k \le m)$를 포함하는 순차 정보의 수가 w_{jk}임을 의미한다.

하나의 군집에 포함되는 순차 정보들에 대한 다중 정렬 방식에 의한 대표 순차패턴 생성 과정에서 가중치 순차패턴 WS의 활용 방법에 대해서는 예제 1에서 보다 명확히 설명한다.

예제 1. (순차 정보 다중 정렬) 하나의 군집 C에 <표 1>에서와 같이 5개의 순차 정보가 포함되고, 이들 순차 정보들을 *밀집*도 내림차순으로 정렬하면 S_3-S_2-S_4-S_5-S_1로 정렬된다. 이때, 이들 순차 정보들은 다음과 같이 정렬된다.

〈표 1〉 하나의 군집에 속하는 순차 정보 및 이들에 대한 다중 정렬

Seq-ID	Sequences	Alignment				
S_1	<(ag)(f)(bc)(ae)(h)>	<(ag)	(f)	(bc)	(ae)	(h)>
S_2	<(ae)(h)(b)(d)>	<(ae)	(h)	(b)	(d)	>
S_3	<(a)(b)(de)>	<(a)		(b)	(de)	>
S_4	<(a)(bcg)(d)>	<(a)		(bcg)	(d)	>
S_5	<(bci)(de)>	<		(bci)	(de)	>
가중치 순차패턴 WS		<(a:4, e:1, g:1):4	(f:1, h:1):2	(b:5, c:3, g:1, I:1):5	(a:1, d:4, e:3):5	(h:1):1> : 5

S_3	<(a)		(b)	(de)>
S_2	<(ae)	(h)	(b)	(d)>
WS_1	<(a:2, e:1) : 2	(h:1) : 1	(b:2) :2	(d:2, e:1) : 2> : **2**

(그림 1) 다중 정렬에 의한 S_3와 S_2의 통합

WS_1	<(a:2, e:1) : 2	(h:1) : 1	(b:2) :2	(d:2, e:1) : 2> : **2**
S_4	<(a)		(bcg)	(d)>
WS_2	<(a:3, e:1) : 3	(h:1) : 1	(b:3, c:1, g:1) :3	(d:3, e:1) : 3> : **3**

(그림 2) 다중 정렬에 의한 WS_1와 S_4의 통합

WS_2	<(a:3, e:1) : 3	(h:1) : 1	(b:3, c:1, g:1) :3	(d:3, e:1) : 3>		: **3**
S_5	<		(bci)	(de)>		
WS_3	<(a:3, e:1) : 3	(h:1) : 1	(b:4, c:2, g:1, i:1) :4	(d:4, e:2) : 4>		: **4**
S_1	<(ag)	(f)	(bc)	(ae)	(h)>	
WS_4	<(a:4, e:1, g:1) : 4	(f:1, h:1) : 2	(b:5, c:3, g:1, i:1) :5	(a:1, d:4, e:3) : 5	(h:1):1 >	: **5**

(그림 3) 나머지 순차 정보 통합

먼저, 순차 정보 S_3과 S_2가 (그림 1)에서와 같이 통합되어 정렬되며, 이들이 통합되어 정렬된 상태는 가중치 순차패턴 WS_1으로 표현된다. 순차 정보 S_3의 첫번째 항목집합 *(a)*와 S_2의 첫번째 항목집합 *(ae)*는 동일한 위치로 정렬된다. 즉, 해당 가중치 순차패턴 WS_1의 첫번째 항목은 *(a:2, e:1):2* 로 표현된다. 이는 두 순차 정보들이 해당 위치를 기준으로 정렬됨을 의미하며, 단위항목 *a*는 2번, *e*는 1번 출현했음을 의미한다. WS_1의 두 번째 위치인 *(h:1):1*은 WS_1의 해당 위치에는 하나의 순차 정보만 정렬됨을 의미하며, 단위항목 *h*는 한 번 출현 했음을 의미한다.

*밀집*도 내림차순으로 정렬된 순차 정보 집합에서 첫번째 단계로 처음 두개의 순차 정보에 대한 통합 정렬 작업이 앞서 기술한 바와 같이 수행되고, 가중치 순차패턴 WS_1을 생성한다. 이어서 다음 순차 정보 S_4와 WS_1을 (그림 2)와 같이 통합한다. 이때, 가중치 순차패턴에 존재하는 하나의 항목집합 $X=(x_1:w_1, \cdots, x_m:w_m):v$와 새롭게 통합 정렬되어야 할 순차 정보에 포함된 하나의 항목집합 $Y=(y_1, \cdots, y_l)$에 대한 갱신 비용 $REPL(X,Y)$는 다음과 같이 구해진다. (아래 식에서 n은 해당 가중치 순차패턴의 전역 가중치를 의미한다.)

$$REPL(X,Y) = \frac{e_R \times v + n - v}{n}$$

$$\text{where } e_R = \frac{\sum_{i=1}^{m} w_i + |Y| \times v - 2\sum_{w_i \in Y} w_i}{\sum_{i=1}^{m} w_i + |Y| \times v} \qquad (5)$$

따라서, $INDEL(X) = REPL(X,0) = 1$이며, $INDEL(Y) = REPL(Y,0) = 1$임을 알 수 있다.

나머지 순차 정보도 밀집도 순서를 고려하여 반복적으로 (그림 3)에서와 같이 통합한다. 이렇게 다섯개의 순차 패턴이 <표 1>에서와 같이 다중 정렬되어 하나의 가중치 순차 패턴 WS_4로 통합된다. 즉, 하나의 군집을 표현하는데 있어서 해당 군집에 속하는 모든 개별 순차 정보를 별도로 유지하지 않고 해당 군집을 표현하는 가중치 순차패턴 하나만 유지하면 된다.

3.2.2 대표 패턴 생성

앞서 기술한 바와 같이 하나의 군집에 속하는 순차 정보들은 다중 정렬 방식에 의한 통합 과정을 통해 하나의 가중치 순차패턴으로 표현된다. 이때 하나의 가중치 순차패턴에서 다수의 순차 정보들에 의해 공유되는 일부분을 추출함으로써 해당 군집에 대한 대표 순차패턴을 생성할 수 있다. 본 절에서는 이러한 대표 순차패턴 생성 과정에 대해서 상세히 기술한다.

하나의 가중치 순차패턴 $WS = <(x_{11}:w_{11}, \ldots, x_{1m_1}:w_{1m_1}):v_1, \ldots, (x_{l1}:w_{l1}, \ldots, x_{lm_l}:w_{lm_l}):v_l>:n$에 대해서 i번째 항목집합에 속하는 단위항목 $x_{ij}:w_{ij}$의 *strength*를 $\frac{w_{ij}}{n} * 100(\%)$로 정의하며, 단위항목의 strength가 클수록 보다 많은 수의 순차 정보들에서 공유되는 단위항목임을 알 수 있다. 이와 같은 사실에 근거하여 strength의 임계값 *min_strength* [0,1]를 설정하여 하나의 가중치 순차패턴으로부터 대표 순차패턴을

생성할 수 있다. 즉, 하나의 가중치 순차패턴에서 *min_strength*보다 작은 strength를 갖는 값들을 제거함으로써 해당 가중치 순차패턴에 대한 대표 순차패턴을 구할 수 있다.

<표 1>의 가중치 순차패턴에서 *min_strength*=30%일 때, 대표 순차패턴은 *<(a)(bc)(de)>*로 구해진다. 한편, 해당 대표 순차패턴을 <표 1>의 순차 정보들과 비교하는 경우 해당 대표 순차패턴은 해당 순차 정보들에서 공유되지만 정확히 포함하는 순차 정보는 존재하지 않음을 알 수 있다. 하지만, 해당 대표 순차패턴에 단 하나의 단위항목을 추가함으로써 S_2를 제외한 다른 순차 정보들과 부분적으로 일치됨을 알 수 있다. 이를 통해서 해당 대표 순차패턴이 통합 대상 순차 정보들에 내재된 공유 정보를 효과적으로 대표할 수 있음을 알 수 있다.

4. 실험 결과

본 절에서는 본 논문에서 제안된 방법의 유용성을 다양한 실험을 통해서 검증한다. 모든 실험들은 듀얼 Xeon-2GB CPU 및 2GB 메인 메모리 사양을 가지며 Red Hat 리눅스 운영체제를 갖는 시스템에서 실험되었다. 논문에서 제안된 방법의 기본적인 성능을 평가하기 위해서 [1]에서 제안된 IBM 데이터 생성기를 통해 생성된 데이터 집합을 활용하였다. IBM 데이터 생성기는 다양한 특성을 갖는 데이터 집합을 효율적으로 생성할 수 있도록 지원하며, 크게 두 가지 과정을 거쳐 데이터 집합을 생성한다. 먼저 사용자에 의해 설정된 매개변수 값을 고려하여 기본 패턴(base pattern)을 임의로 생성하고, 이어서 이들 기본 패턴들을 다양하게 조합하여 순차 데이터 집합을 생성한다. 따라서 기본 패턴은 해당 순차 데이터 집합을 구성하는 다수의 순차 정보들에서 공유되는 대표 패턴이라 간주할 수 있으며, 이러한 기본 패턴들을 효과적으로 탐색하는 것이 순차패턴 마이닝의 목표가 될 수 있다. 본 절의 실험에서 사용된 데이터 집합 생성을 위한 IBM 데이터 생성기의 매개변수 및 설정 값은 <표 2>에서와 같다.

〈표 2〉 데이터 생성 및 마이닝을 위한 매개변수 및 기본 설정 값

변 수	의 미	비교실험 설정값	기본실험 설정값
$\|I\|$	단위항목의 수	100	1000
N_{seq}	순차 정보의 수	1000	10000
N_{pat}	기본 순차패턴의 수	10	100
L_{seq}	순차 정보의 평균 항목집합의 수	10	20
L_{pat}	기본 순차패턴의 평균 항목집합의 수	7	14
I_{seq}	순차 정보의 항목집합에서 평균 단위항목의 수	2.5	2.5
I_{pat}	기본 순차패턴의 항목집합에서 평균 단위항목의 수	2	2
k	군집화 매개변수 : 근린 순차 정보의 수	6	5
min_strength	대표 패턴 생성을 위한 strength 임계값	50 %	50 %

4.1 유용성 검증 척도

기존의 일반적인 순차패턴 탐색 방법들은 길이가 짧고 평범한 방대한 수의 순차패턴을 마이닝 결과로 구하는 반면 본 논문에서 제안하는 근사 순차패턴 탐색 방법은 의미적으로 중요하면서 충분히 긴 순차패턴을 마이닝 결과로 구해준다. 이때, 마이닝 결과 집합의 유용성을 평가하기 위한 척도로서 *기본 패턴 탐색율*을 제안한다. 기본 패턴 B와 마이닝 결과 구해진 대표 패턴 P에 대해서 $B \otimes P$를 두 패턴들에서 공통되는 최대 부분패턴이라 할 때, ***기본 패턴 탐색율*** R은 IBM 데이터 생성기에서 데이터 집합 생성시 이용되는 기본 패턴들 중에서 마이닝 결과 집합으로 구해지는 패턴의 비율을 나타내는 것으로서 다음과 같이 정의한다.

$$R = \sum_{base\ pat\ B} \{E(F_B) \times \min\{1, \frac{\max_{con\ pat\ P}(|B \otimes P|)}{E(L_B)}\}\} \quad (5)$$

여기서, $E(F_B)$ 및 $E(L_B)$는 기본 패턴의 출현빈도 수 기대값 및 패턴 길이의 기대값을 나타내며, 데이터 집합 생성시 IBM 데이타 생성기에서 설정된다. $E(L_B)$값은 데이터 집합 생성시 설정되는 기대값으로서 경우에 따라서는 실제 값 $|B \otimes P|$가 $E(L_B)$보다 더 커질 수 있다. 이런 경우 해당 기본 패턴의 기본 패턴 탐색율을 100%로 간주할 수 있으며 $\frac{\max_{con\ pat\ P}(|B \otimes P|)}{E(L_B)}$ 값이 100%보다 커지는 경우 100%로 간주한다. 이를 통해 기본 패턴 탐색율은 0% 이상 100% 이하의 값을 유지하게 된다. 하나의 마이닝 결과 집합에 대해서 기본 패턴 탐색율이 클수록 데이터 집합 생성시 사용된 기본 패턴들을 보다 잘 탐색하는 것으로 판단할 수 있다.

4.2 비교 실험

본 절에서는 본 논문에서 제안된 방법의 효율성을 기존의 접근 방법과 비교하기 위한 실험 결과를 기술한다. 본 논문에서 제안된 근사 순차 패턴 탐색 방법의 효율성을 기존의 지지도 기반의 순차 패턴 방법들과 비교하여 평가하기 위한 방법으로 지지도 기반 방법들에서 얻어지는 마이닝 결과와 본 논문에서 제안된 방법으로 얻어지는 마이닝 결과를 비교하였다. 일반적으로 지지도 기반 순차 패턴 탐색 방법들은 서로 동일한 결과를 구해준다. 따라서 지지도 기반 순차 패턴 탐색 결과를 구하기 위한 마이닝 방법의 종류에 무관하게 동일한 결과를 얻을 수 있으며, 본 논문에서는 [2]에서 제안된 방법을 이용하여 지지도 기반 순차 패턴 탐색 결과를 구했다. 비교 실험을 위해서 <표 2>의 비교실험 설정값에서와 같이 10개의 기본 패턴으로부터 생성된 1000개의 순차 정보로 구성되는 데이터 집합을 생성하였다. 생성된 데이

〈표 3〉 비교 실험 결과

접근방법	근사 패턴	지지도 기반
매개변수 설정값	k = 6	min_sup = 6%
기본 패턴 탐색율	91.16 %	92.52 %
탐색된 패턴 총수	8	128936
허위 패턴 수	0	16
중복 패턴 수	1	128910

터 집합에 대해서 본 논문에서 제안된 방법과 지지도 기반 순차 패턴 탐색 방법을 적용하여 마이닝 결과를 구한 결과는 <표 3>에서와 같다. 표에서 '탐색된 패턴 총 수'는 수행 결과 얻어진 패턴의 총 수를 의미한다. 허위 패턴은 수행 결과 얻어진 패턴 중에서 해당 허위 패턴과 가장 유사한 기본 패턴을 찾아서 비교한 결과 일치되는 단위항목의 수보다 불일치되는 단위항목의 수가 많은 경우를 지칭하며, '허위 패턴 수'는 이러한 패턴의 수를 의미한다. 중복 패턴은 수행 결과 얻어진 패턴들을 기본 패턴들과 비교(일치되는 단위항목들을 비교한 결과)한 결과 동일한 기본패턴으로부터 파생된 것으로 판단되는 두 개 이상의 패턴을 지칭하며, '중복 패턴의 수'는 이러한 패턴의 수를 의미한다. 표에서 보는 바와 같이 기본 패턴 탐색율은 두 접근 방법 모두 90% 이상으로 나타난다. 하지만, 기존의 지지도 기반 방법은 마이닝 결과 얻어지는 결과 패턴의 수가 지나치게 많다. 즉, 1000개의 순차 정보를 분석한 결과 12만개 이상의 결과 패턴을 얻게 된다. 이들 대부분은 서로 유사한 패턴들이 중복된 것으로서, 이러한 분석 결과는 해당 분석 결과를 활용하는데 있어서 효용성을 떨어뜨린다. 반면, 본 논문에서 제안된 방법은 총 8개의 대표 패턴을 탐색하며, 그 중 하나는 중복 패턴이다. 즉, 1000개의 순차 정보를 8개의 기본 패턴으로 효율적으로 축약하면서도 기본 패턴 탐색율은 90%이상을 나타낸다. 한편, 기본적인 접근 방법이 다른 마이닝 방법들간의 비교에 있어서 그 밖의 다른 성능 비교는 불필요한 것으로 판단되어 본 논문에서는 생략한다.

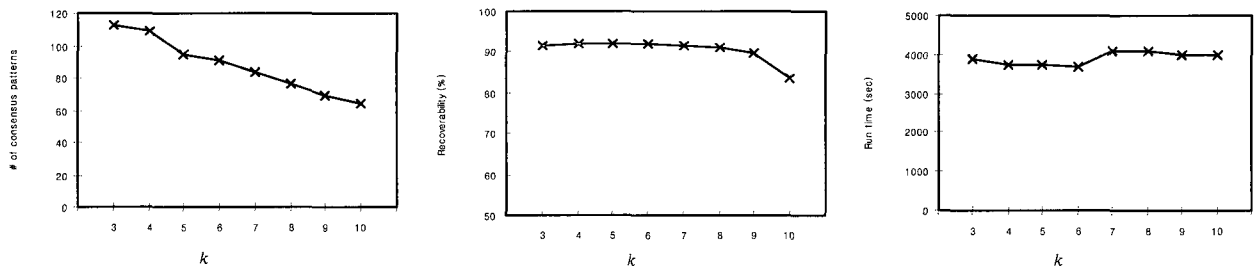
4.3 기본 성능 평가

본 절에서는 데이터 생성을 위한 매개변수 값을 <표 2>의 기본실험 설정값에서와 같이 설정하여 생성된 데이터 집합을 이용하여 본 논문에서 제안된 근사 순차패턴 탐색 방법의 성능을 평가하였다. IBM 데이터 생성기에 의해 생성된 데이터 집합들은 기본 패턴을 명확히 파악할 수 있으므로, 본 논문에서 제안된 근사 순차 패턴 탐색 방법의 성능을 4.1절에서 기술된 척도 등에 근거하여 명확히 평가할 수 있도록 지원한다.

실험은 크게 두가지로 구분된다. 하나는 논문에서 제안된 방법의 성능에 영향을 미치는 중요 매개 변수 (군집화 매개변수 k)에 대하여 해당 매개변수의 변화에 따른 제안된 방법의 성능 변화를 분석하였다. 다른 하나는 분석 대상 순차 데이터베이스의 순차 정보 개수 변화(즉, 분석 대상 데이터 집합의 규모 변화)에 따른 제안된 방법의 성능 변화 실험이다.

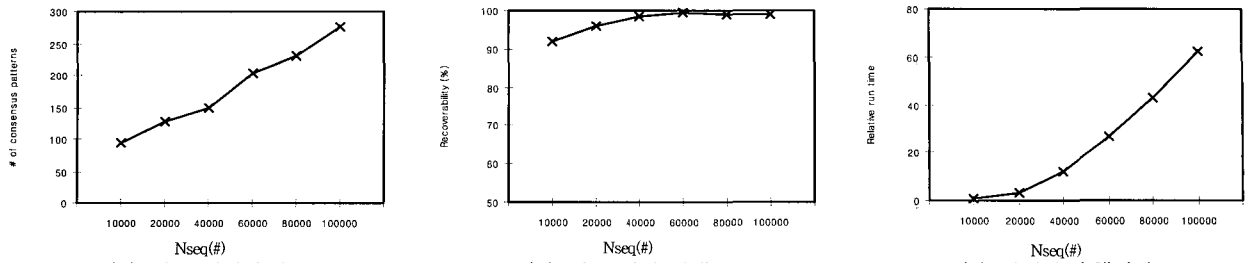
(그림 4)는 다른 실험 조건들은 <표 2>의 기본실험 설정값에서와 동일하게 설정하고 군집화 매개변수 k 값을 다양하게 변화시켰을 때 본 논문에서 제안된 근사 순차패턴 탐색 방법의 성능 변화를 보여준다. 일반적으로 k-NN 군집화 방법에서는 k 값이 증가될수록 생성되는 군집의 수가 감소된다. 논문에서 제안된 방법에서는 순차 정보에 대한 군집화 과정에서 생성된 개별 군집당 하나의 대표 패턴을 생성한다. 따라서, (그림 4) (a)에서 보듯이 k 값이 증가됨에 따라 대표 순차패턴의 수가 감소된다. 대표 순차패턴 수가 감소함에 따라 기본 패턴 탐색율도 다소 감소하지만, 많은 경우 큰 영향을 미치지 않는다. (그림 4) (b)에서 보듯이 k 값이 증가하더라도 탐색율은 거의 변화없이 유지되거나 약간 감소된다. 특히 그림에서 알 수 있듯이 $3 \le k \le 9$ 범위일 경우 탐색율에 거의 변화가 없고, k가 10이 되었을 때 비로소 탐색율이 감소 됨을 알 수 있다. 즉, 본 실험에서 최상의 결과를 주는 k 값의 범위는 $3 \le k \le 9$이며, 본 논문에서 제안된 알고리즘이 k 값의 변화에 크게 영향을 받지 않음을 알 수 있다. 이는 k 값이 증가할 때 감소되는 대표 패턴들은 대부분 중복 패턴들이기 때문에 탐색율에는 거의 영향을 미치지 않기 때문이다. (그림 4) (c)는 수행 시간을 보여주며, k 값이 증가됨에 따라 수행 시간이 약간은 증가되나, 크게 영향을 받지 않음을 알 수 있다.

(그림 5)는 다른 실험 조건들은 <표 2>의 기본실험 설정값에서와 동일하게 설정하고 순차 정보의 수를(데이터 집합의 크기) 다양하게 변화시켰을 때 본 논문에서 제안된 근사 순차패턴 탐색 방법의 성능 변화를 보여준다. 일반적으로 순차 정보의 수가 많을수록 생성되는 군집의 수가 증가된다. 따라서, (그림 5) (a)에서 보듯이 순차 정보의 수가 증가됨에 따라 대표 순차패턴의 수가 증가된다. 한편, (그림 5) (b)에서 보듯이 순차 정보의 수가 증가될수록 기본 패턴 탐색율은 증가된다. 이것은 동일한 개수의 기본 패턴에서 생성되는 데이터 집합에 있어서는 데이터 집합의 규모가 증가될수록 해당 기본 패턴들이 해당 데이터 집합에 보다 많이, 보다 명확히 표현되며, 따라서 기본 패턴 탐색율이 증가된다. (그림 5) (c)는 제안된 방법의 상대적 수행 시간을 보여준다. 순차 정보의 수가 증가될수록 군집화 과정이 길어지고 대표 순차패턴을 탐색해야 하는 군집의 수가 증감되므로 수행 시간이 비례적으로 증가된다.



(a) 대표 패턴의 수 (b) 기본 패턴 탐색율 (c) 수행시간

(그림 4) k 값 변화에 따른 성능 변화



(a) 대표 패턴의 수 (b) 기본 패턴 탐색율 (c) 상대적 수행시간

(그림 5) 분석 대상 순차 정보의 수 증가에 따른 성능 변화

5. 결론 및 향후 연구 방향

순차 패턴 탐색을 활용하고자 하는 다수의 응용 분야에서는 정확한 일치에 기반하여 탐색된 방대한 수의 평범한 패턴들을 필요로 하기 보다는 의미적으로 중요하면서도 보다 긴 패턴을 분석 결과로 얻을 수 있기를 기대하며, 이러한 순차패턴들은 짧은 길이의 평범한 순차패턴들에 비해 해당 응용 분야에서 훨씬 유용하게 활용 될 수 있다.

본 논문에서는 이러한 사실에 근거하여 데이터베이스에 대한 근사 순차패턴 탐색 방법을 제안하였다. 근사 순차패턴은 분석 대상이 되는 순차 데이터베이스에 속하는 다수의 순차 정보들에서 근사적으로 공유되는 순차패턴으로서, 기존의 정확한 일치에 기반하여 탐색된 순차패턴들에 비해 일반적으로 길고 유용한 순차패턴으로 탐색한다. 대용량 데이터 집합에서 근사 순차패턴을 탐색하기 위해서 본 논문에서 제안된 방법에서는 순차 정보들에 대한 군집화 과정을 수행하고, 개별 군집에 포함된 순차 정보들을 다중 정렬화 방법을 이용하여 하나로 통합하여 가중치 순차패턴 형태로 표현하며 이로부터 대표패턴을 생성한다. 성능 검증 실험을 통해 본 논문에서 제안된 방법이 데이터 생성시 활용된 기본 패턴들을 효율적으로 탐색함을 확인하였다.

한편, 본 논문에서 제안하는 근사 순차 패턴 탐색 방법은 전체 수행 과정에서 순차 정보에 대한 군집화 과정에 상대적으로 많은 시간이 할애된다. 따라서, 보다 효율적인 군집화 방법 등을 적용하는 경우 수행 시간 등의 성능을 향상시킬 수 있을 것으로 판단되며, 이를 포함한 근사 순차 패턴 탐색 방법의 성능 개선 문제는 향후 의미있게 다뤄볼 수 있는 연구 주제가 될 것으로 판단된다.

참 고 문 헌

[1] R. Agrawal and R. Srikant, Mining Sequential Patterns, *Proceedings of the 11th Int'l Conference on Data Engineering*, pp.3-14, Taipei, Taiwan, Mar., 1995.

[2] J. Pei, J. Han, B. Mortazavi-Asi, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, PrefixSpan : Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, *Proceedings of the 17th Int'l Conference on Data Engineering*, 2001.

[3] X. Yan, J. Han, and R. Afshar. CloSpan : Mining Closed Sequential Patterns in Large Datasets. In *Third SIAM International Conference on Data Mining (SDM)*, pp.166-177, San Francisco. CA, 2003.

[4] J. Ayres, J. Flannick, J. Gehrke, and T. Yiu. Sequential pattern mining using a bitmap representation. *Proceedings of the ACM SIGKDD Int'l Conferences on Knowledge Discovery and Data Mining*, pp.429-435, Edmonton, Canada, Jul., 2002.

[5] S. Parthasarathy, M.J. Zaki, M. Ogihara, and S. Dwarkadas, Incremental and Interactive Sequence Mining, *Proceedings of the 8th Int'l Conference on Information and Knowledge Management*, 1999.

[6] O. Gotoh. Multiple sequence alignment: Algorithms and applications. *Advanced Biophysics*, Vol.36, pp.159-206, 1999.

[7] D. Gusfield. Algorithms on strings, trees, and sequences. *Computer Science and Computational Biology*, Cambridge University Press, Cambridge, England, 1997.

[8] J. Thompson, F. Plewniak, and O. Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*. Vol.27, No.13, pp.2682-2690, Oxford University Press. 1999.

[9] C. Yang, U. Fayyad, and P.S. Bradley. Efficient discovery of error-tolerant frequent itemsets in high dimensions. *Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pp.194-203, 2001.

[10] J. Yang, P. S. Yu, W. Wang, and J. Han. Mining long sequential patterns in a noisy environment. In *Proc. of ACM Int'l Conference On Management of Data (SIGMOD)*, pp.406-417, Madison, WI, June, 2002.

[11] M. A. Wong and T. Lane. A kth Nearest Neighbor Clustering Procedure. In *Journal of the Royal Statistical Society*, Series B, 45, pp.362-368, 1983.

금 혜 정

e-mail : kum@email.unc.edu
1995년 연세대학교 컴퓨터과학과(학사)
1997년 University of North Carolina, Chapel Hill(석사)
2004년 University of North Carolina, Chapel Hill(박사)
2004년~현재 University of North Carolina, Chapel Hill 연구교수
관심분야 : 데이터 마이닝, 생물정보학, 다중 데이터 분석, 패턴 매칭, 사회적 네트워 분석

장 중 혁

e-mail : jhchang@amadeus.yonsei.ac.kr
1996년 연세대학교 컴퓨터과학과(학사)
1998년 연세대학교 대학원 컴퓨터과학과 (석사)
2005년 연세대학교 대학원 컴퓨터과학과 (박사)
2005년~현재 연세대학교 소프트웨어응용 연구소 전문연구원
관심분야 : 데이터 스트림, 데이터 마이닝, 정보보안, 생물정보학