제 19 회 한국정보처리학회 춘계학술발표대회 논문집 제 10 권 제 1 호 (2003. 5)

# 순차 패턴 알고리즘의 분류 및 분석*

이양우, 이헌규, 김룡, 서성보, 류근호
충북대학교 데이터베이스 연구실
e-mail : {dooji, sbseo, khryu}@dblab.chungbuk.ac.kr

# Classification and Analysis of Sequential Pattern Algorithms

Yang-Woo Lee, Hohn Gyu Lee, Lyong Kim, Sung Bo Seo, Keun Ho Ryu
Database Laboratory, Chungbuk National University

요 약

순차 패턴 마이닝은 대량의 시퀀스 데이터베이스에서 빈발 서브 시퀀스를 찾는 기법이다. 지금까지 많은 순차 패턴 마이닝에 관한 연구들이 순차 패턴을 효율적으로 찾기 위하여 제안되었다. 그러나 제안된 방법들은 응용에 적용할 수 있도록 체계적으로 분류되어 있지 않다. 따라서 이 논문에서는 알고리즘에 대한 연구들을 분류하고 이들 중 대표적인 알고리즘들을 선정하여 각각에 대해 분석하였다. 그리고 각 응용 도메인에 적용한 연구들과 기술적인 문제를 해결하는 연구들에 대해 정리하였다. 마지막으로 성능 향상을 위한 기법이나 자료 구조에 대해 언급하고 향후 순차 패턴 마이닝의 연구 방향을 제시하였다. 이 연구는 실제 응용에 적합한 순차 패턴 마이닝 알고리즘의 선택과 향후 새로운 순차 패턴 알고리즘 연구의 기반을 제공할 것이다.

## 1. 서론

시퀀스 데이터베이스에서 빈발 서브 시퀀스를 찾는 순차 패턴 마이닝은 클러스터링, 분류 등과 함께 데이터 마이닝에서 중요한 기법 중 하나이다. 이 기법은 1995년 Agrawal[1]이 제안한 지 10년이 되지 않았지만, 현재 많은 분야에서 연구되어 여러 가지 효율적이고 새로운 기법들이 제안되고 있다. 그 동안 알고리즘에 대한 연구도 있었고, 실제 응용 도메인에 적용한 연구도 있었다. 이런 순차 패턴 기법으로 얻은 지식은 소비자의 구매 순서나 질병의 발병 순서와 같은 일련의 순서를 갖는 데이터에 적용되어 소비자의 구매를 돕거나 상품 마케팅에 활용되고 환자의 병을 예측하여 미리 대응하거나 환자의 병에 대한 의사 결정에 도움을 주고 있다. 최근에 각광 받고 있는 바이오 분야의 DNA 시퀀스 분석에 적용되고 있다.

초기 순차 패턴 알고리즘의 경향은 처음에 제안된 Agrawal의 알고리즘[1]에 기초한 것들이 대부분이었으나 2000년에 후보 집합을 생성하지 않는 연관 규칙 알고리즘인 FP-tree기법이 소개된 후에 이를 응용한 FreeSpan[10]이라는 순차 패턴 알고리즘이 나왔다. 지금까지 많은 연구들이 있었지만, 이들에 대한 체계적인 분류는 고려하지 않았다. 여기서는 기존의 많은 연구를 분류하고 각 분류에서 대표적인 알고리즘을 선정하고 각각을 분석한다. 또한 알고리즘을 응용 도메인에 적용한 연구들과 기술적인 문제를 해결하려는 연구들에 대해서도 정리한다. 이런 연구를 통해 실제 응용에 적합한 순차 패턴 마이닝 알고리즘의 선택과 새로운 순차 패턴 알고리즘에 대한 연구의 기반이 될 것이다.

이 논문에서는 먼저 관련 연구로서 순차 패턴 마이닝에 대한 기본적인 개념을 소개하고 순차 패턴 알고리즘 연구에 대해 3 가지로 분류한다. 그리고 각 분류에 속하는 알고리즘 중 대표적인 것을 선정하여 각각을 분석하고 그 특성을 설명한다. 응용 도메인과 기법에 대해 설명하고 이들의 특징에 대해서도 언급한다. 마지막으로 성능 향상을 위한 기법과 알고리즘에 적용할 수 있는 자료 구조에 대해 언급하고 향후 순차 패턴 마이닝의 연구 방향을 제시한다.

## 2. 순차 패턴 마이닝 및 알고리즘의 분류 기준

### 2.1 순차 패턴 마이닝

1995년 Agrawal이 연관 규칙을 기반으로 순차 패턴 마이닝 기법[1]을 제안하였다. 이 기법은 연관 규칙과 달리 시간 순서를 갖는 시퀀스 데이터를 다루고 찾아진 결과도 빈발한 서브 시퀀스이다. 고객의 구매 패턴에서 자연 재해, 주식 가격의 변화, 질병에 관한 데이터나 DNA 시퀀스에 이르기까지 대부분의 데이터가 시간 순서를 갖는 데이터로서 순차 패턴 마이닝의 대상이 된다.

순차 패턴 마이닝 기법은 항목 사이의 연관성을 측정하는 연관 규칙(association rule)에 순서를 고려하여 유용한 지식을 찾는 기법이라고 정의할 수 있다. 그러므로 연관 규칙과 달리 각 항목마다 시간 정보가 포함되어 있어야 하고 모든 규칙은 순서적으로 나열되어야 한다. 예를 들어, 연관 규칙에서 항목 A, B에 대하여 A가 먼저 발생했다고 가정할 때, A→B, B→A라는 규칙이 발견될 수 있지만 순차 패턴에서는 A→B라는 규칙만이 발견될 수 있다.

사전 항목들의 전체 집합을 $I=\{i_1, i_2, \ldots, i_m\}$이라 하고, I에 대한 두 개의 부분 집합 X, Y가 있을 때, 패턴은 "X ➔ Y"

* 이 연구는 한국과학재단 지정 청주대 RRC(정보통신 연구센터)의 지원으로 수행되었음.

로 표시된다. 이 때, X와 Y는 서로 소(X ∩ Y=Ø)라고 가정한다. 이 패턴은 "X에 속한 항목들이 발생이 Y에 속한 항목들의 발생을 야기한다."라고 해석한다.

이렇게 얻어진 패턴의 유용성을 측정하기 위해 순차 패턴 마이닝은 연관 규칙과 마찬가지로 지지도(support)와 신뢰도(confidence)를 이용한다. 전체 데이터베이스를 T라하고 항목 A와 B가 있을 때, 지지도는 전체 트랜잭션 T에서 항목 A와 B를 포함하는 트랜잭션의 비율로 구한다. 이는 해당 패턴의 연관성을 위해 최소한 확보해야 할 통계적인 중요도이다. 신뢰도는 항목 A를 포함하는 모든 트랜잭션 중에서 항목 B까지 포함하는 트랜잭션의 비율로 구한다.

### 2.2 분류 기준

순차 패턴 알고리즘에 대한 많은 연구를 통해 순차 패턴 알고리즘은 다음과 같이 세 가지로 분류할 수 있다. 또한 제약을 포함하는 알고리즘은 따로 표시(*)로 구분하였다.

<표 1> 순차 패턴 알고리즘의 분류

| 분류 | 년도 | 알고리즘 |
|---|---|---|
| Apriori-like | 1995 | AprioriAll et al.[1] |
| | 1996 | GSP[2] *† |
| | 1998 | Max-Miner[3] |
| | 1998 | PSP[4] |
| | 1999 | SPIRIT[5,6] * |
| | 2002 | |
| | 1998 | SPADE[7,8] |
| | 2001 | |
| | 2000 | cSPADE[9] * |
| Pattern-growth | 2000 | FreeSpan[10] |
| | 2001 | PrefixSpan[11] |
| | 2002 | DELISP[12] |
| Others | 1995 | MSDD[13] |
| | 1995 | WINEPI[14,15] |
| | 1997 | |
| | 1999 | pSPADE[16] |
| | 2000 | EVE[17] |
| | 2002 | webSPADE[18] |

#### 2.2.1 Apriori-like 알고리즘

이전의 결과를 다음의 수행에 이용하는 방법으로 후보 집합을 생성하고 임계치를 만족하지 못하는 것을 제거한 후 빈발 집합을 만든다. 이 빈발 집합은 다시 조인 연산으로 다음 후보 집합을 생성한다. 이런 과정의 반복으로 수행되며, 더 이상 후보 집합이 생성되지 않으면 멈추게 된다.

알고리즘은 단순하지만, 반복적으로 대량의 후보 집합을 생성하고 데이터베이스를 여러 번 읽는 문제와 길이가 긴 순차 패턴 마이닝이 매우 어렵다는 문제를 갖는다.

#### 2.2.2 Pattern-growth 알고리즘

마이닝을 하는 작업과 데이터베이스를 분해하기 위해 분할-정복 기법을 이용하여 문제를 해결하였다. 데이터베이스를 여러 번 읽는 문제를 해결하기 위해 프로젝션 데이터베이스를 만들어 검색 공간의 크기를 점차 줄인다. 또 후보 집합 생성 및 테스트 비용을 없애기 위해 패턴-조각 확장(pattern-fragment growth) 기법을 사용했다.

#### 2.2.3 기타 알고리즘

여기에는 네트워크에서 발생하는 이벤트와 병렬화에 관한 알고리즘을 기타 알고리즘으로 분류하고 분석하였다.

## 3. 알고리즘의 분류 및 분석

### 3.1 Apriori-like 알고리즘

#### 3.1.1 AprioriAll, AprioriSome, DynamicSome 알고리즘[1]

이 알고리즘들은 [1]에서 제안한 기법으로 기존의 연관 규칙 알고리즘에 순서를 추가하여 대량의 고객 트랜잭션이 포함된 데이터베이스에서 순차 패턴을 찾는다. 이 알고리즘은 시장 바구니 분석을 위해서 고안되었고 각 트랜잭션은 고객-id, 트랜잭션-시간, 트랜잭션에서 구입한 아이템들로 이루어진다. 이 알고리즘은 고객-id와 트랜잭션-시간에 따라 정렬하는 단계, 빈발 아이템 셋을 생성하는 단계, 각 트랜잭션을 빈발 아이템 셋의 집합으로 변환하는 단계, 원하는 시퀀스를 찾는 단계, 찾아진 시퀀스에서 최대인 것을 찾는 단계로 구분된다.

먼저 AprioriAll기법은 모든 길이의 빈발 아이템 셋을 찾아서 결과를 준다. AprioriSome기법은 특정 길이의 빈발 아이템 셋을 선택적으로 찾는다. Backward절을 이용하여 건너뛴 길이의 후보 아이템 셋을 찾는다. 마지막으로 Dynamic기법은 사용자가 정의한 step의 값을 이용하여 띄엄띄엄 빈발 아이템 셋을 생성하고, 더 이상 없으면 이전의 빈발 아이템 셋을 찾는 알고리즘이다.

실험 결과는 *AprioriAll*은 *AprioriSome*보다 수행 속도는 느리지만, 적은 메인 메모리를 가지고 많은 잠재적인 빈발 시퀀스를 찾을 때에는 이점을 가지고 있다는 것을 보여준다.

#### 3.1.2 GSP 알고리즘[2]

GSP(Generalized Sequential Patterns)는 [1]을 제안했던 저자들이 확장한 알고리즘이다. 확장된 세가지 부분 중 첫 번째는 순차 패턴의 이웃한 원소 사이의 최대 최소 갭(gap)이라는 시간 제약을 명시하여 최소보다 작으면 동시에 발생한 것으로 하고 그렇지 않으면 별개의 시퀀스로 분리한다. 다음은 순차 패턴의 한 원소 내의 처음 이벤트와 마지막 이벤트 사이의 최대 시간(sliding window)을 명시하여 최대 시간보다 다른 패턴으로 분리한다. 마지막으로 정적인 사용자 정의 분류(static taxonomy)가 주어졌을 때, 서로 다른 레벨을 교차하여 순차 패턴이 가능하게 하는 것이다.

이런 제약은 더 적은 후보 집합을 생성하여 실행 시간을 단축시킨다. 데이터베이스의 크기가 같을 때, 데이터 시퀀스의 아이템들의 수에 따라 GSP의 실행 시간이 서서히 증가하는 경향이 있다.

#### 3.1.3 Max-Miner 알고리즘[3]

Apriori 기법을 이용한 알고리즘들의 단점 중 하나는 긴 순차 패턴을 찾는 것이 어렵다는 것이다. Max-Miner기법은 이런 문제를 해결하기 위해 제안된 알고리즘으로 유전자 데이터베이스와 같이 긴 패턴을 찾는 곳에서 유용하다. 이 알고리즘은 긴 빈발 아이템 셋을 빠르게 찾기 위해 "look ahead"를 시도한다. 앞서서 긴 빈발 아이템 셋을 확인하여 모든 서브 셋을 줄일 수 있다.

이 알고리즘과 유사한 것으로 MaxFclt와 MaxClique알고리즘이 있는데, 이는 검색 전체에서 "look ahead"를 하는 것이 아니라 초기화하는 동안에만 이 기법을 사용한다. 또 Pincer-Search 알고리즘은 단지 빈발하지 않은 아이템 셋을 포함하는 긴 후보 아이템 셋이 없다는 것을 보증하는 알고리즘이다. 이 논문의 마지막에서 소개한 Max-Miner-LO(Longest Only)는 오직 가장 긴 최대 빈발 아이템을 찾기 위한 알고리즘이다.

실험 결과는 긴 빈발 패턴을 찾을 때, Apriori 알고리즘과 Max-Miner 알고리즘의 큰 차이를 보여준다.

#### 3.1.4 PSP 알고리즘[4]

PSP(Prefix Tree For Sequential Patterns) 알고리즘은 GSP 알

† *는 constraint 를 갖는 알고리즘

고리즘과 거의 흡사하다. 단 GSP보다 좀더 효율적으로 순차 패턴 마이닝을 하기 위해 새로운 중간 데이터 구조를 만들어 이용한다.

GSP알고리즘이 해시 트리를 이용하는데 반해 PSP알고리즘은 prefix 트리를 이용하며, 단 노드는 루트에서 단 노드까지의 시퀀스의 지지도를 포함한다. 그러므로 이 알고리즘은 공통의 prefix를 갖는 요소를 갖는 후보 시퀀스들로 연결되어 있어 모든 후보 시퀀스를 저장하는 GSP알고리즘보다 적은 메모리가 요구된다.

3.1.5 SPIRIT 알고리즘[5]

기존의 순차 패턴 마이닝이 대량의 데이터베이스에서 유용한 패턴을 찾기 위해 최소 지지도와 같은 한정된 제약을 이용하였다. SPIRIT(Sequential Pattern Mining with Regular Expression Constraints) 알고리즘에서는 사용자가 원하는 데이터를 찾기 위해 여러 가지 제약을 명시할 수 있도록 한다. 이를 위해 Regular Expression(RE)를 사용한 SPIRIT 알고리즘을 제안하였다. 이런 제약들을 이용해 더 많은 후보 집합을 가지치기함으로써 좀더 정확하고 간결한 순차 패턴을 얻도록 한다. 이는 사용자의 관심에 초점을 맞춘 기법이지만, 결과적으로 적은 후보 집합을 생성함으로써 실행 시간이 빨라지고 긴 빈발 패턴에 대한 이해가 쉽게 되었다. 그러나 자칫 사용자의 관심에 너무 많은 초점을 두면, 엄청난 계산 비용을 초래할 수 있기 때문에 이 둘 사이의 적절한 조절이 필요하다.

이런 제약을 위해 고려해야 할 것이 사용자가 관심을 갖는 순차 패턴을 표현할 수 있는 유연한 제약 조건의 표현 언어와 사용자가 명시한 제약 조건을 마이닝 과정에 적절히 포함시키기 위한 방법이다. 위에서 언급했듯이 여기서는 RE를 이용한다. 이는 자연적인 구문과 넓은 범위의 흥미롭고 중요한 제약을 표현하는데 충분한 표현력을 가진다. 또 제약을 주는 정도에 따라 다음의 4가지로 분류하여 각각을 설명하고 있다.

2002년도에 이것을 다시 정리한 논문[6]이 제출되었다.

3.1.6 cSPADE 알고리즘[9]

cSPADE(Constrained Sequential Pattern Discovery using Equivalence classes)는 1998년에 제안한 SPADE알고리즘[7]에 시퀀스의 길이, 너비, 시간 제한과 이벤트 제약, 클래스 정보 제약을 포함한 알고리즘이다. 주로 GSP알고리즘과의 비교하고 있다. 특히 이 알고리즘이 더 구현하기 쉽고 성능 평가를 통해 더 효과적인 해결책임을 증명하였다.

GSP가 수평(horizontal) 데이터베이스를 사용하는데 반해 cSPADE는 수직(vertical) 데이터베이스를 사용하는 것이 이 알고리즘의 특징이다. 전자가 테이블의 모든 행이 하나의 트랜잭션임에 반해 후자는 각 아이템이 고객-id와 트랜잭션-시간 쌍의 리스트를 갖는다.

그러나 이 알고리즘은 많은 메모리를 요구하므로 메인 메모리가 비교적 크고 아이템의 수가 클 때 유용하다. 또한 2001년에는 SPADE[8]를 상세히 정리한 논문이 제출되었다.

### 3.2 Pattern-growth 알고리즘

3.2.1 FreeSpan 알고리즘[10]

FreeSpan(Frequent Pattern-projected Sequential Pattern Mining) 알고리즘은 데이터베이스를 빈발 서브 패턴의 발생 가능성을 고려하여 반복적으로 시퀀스 데이터베이스를 프로젝트(project)하여 작게 프로젝트 된 데이터베이스 집합을 만들고 각 프로젝트 된 데이터베이스에서 서브 시퀀스 조각들을 확장시켜 빈발 패턴을 찾는다.

이 알고리즘은 데이터베이스를 한번 스캔하여 찾아진 빈발 아이템을 빈발 횟수의 내림차순으로 정렬하여 빈발 아이템 리스트를 만들고 이 아이템들을 가로축과 세로축에 놓아 빈발 아이템 행렬을 만든다. 다음으로 길이가 2인 빈발 패턴을 찾고 프로젝트 된 데이터베이스와 반복되는 아이템에 표시를 한다. 다시 반복되는 아이템 패턴과 프로젝트 데이터베이스를 생성한다. 여전히 탐사할 패턴이 있다면, 프로젝트 된 데이터베이스에서 행렬 프로젝션 탐사를 반복적으로 한다. 이 알고리즘은 GSP알고리즘에 비해 대체로 적은 수의 서브 시퀀스 결합으로 빠른 성능을 갖는다.

3.2.2 PrefixSpan 알고리즘[11]

PrefixSpan(Prefix projected Sequential pattern mining) 알고리즘은 앞서 소개한 FreeSpan 알고리즘을 개선한 것이다. FreeSpan 알고리즘의 검색 대상이 시퀀스 데이터베이스를 프로젝션 하는 것이라면, 이 알고리즘은 prefix의 확장을 통해 빈발 패턴을 발견한다. 이 방법은 단순히 prefix 서브 시퀀스를 평가하고 이들에 대응하는 postfix 서브 시퀀스만을 프로젝트 함으로 효율성이 좋다.

이 연구는 먼저 level-by-level projection 기법을 설명하고 이를 보완하여 bi-level기법을 소개하였다. 두 기법을 비교한 결과를 보면, level-by-level기법의 프로젝션 수가 bi-level기법의 프로젝션의 수보다 두 배가 넘게 발생한다. 이는 가장 많은 비용을 차지하는 프로젝션을 획기적으로 줄였다는데 의의가 있다.

이 연구에서 또 하나 주목할 점은 메모리의 효율성을 위해 의사-프로젝션(pseudo-projection)을 제안했다는 것이다. 한 시퀀스에 대해 각기 다른 prefix를 갖는 시퀀스의 postfix를 각각 저장한다면 postfix의 중복이 생기게 된다. 이런 중복을 줄이기 위해 시퀀스의 이름을 가리키는 포인터와 postfix의 시작 위치의 쌍으로 표시한다.

이 알고리즘의 장점은 후보 집합을 생성하지 않는다는 것과 프로젝트 된 데이터베이스가 초기 데이터베이스에 비해 작아진다는 것 그리고 가장 많은 비용이 요구되는 프로젝트 데이터베이스 생성의 비용을 줄이기 위해 bi-level 프로젝션 방법을 이용했다는 것이다. 단점이자 향후 연구로서 GSP에서 제안한 시간 제약 조건을 포함하도록 확장하는 것을 제시하였다.

### 3.3 기타 알고리즘

3.3.1 MSDD 알고리즘[13]

MSDD(Multi-Stream Dependency Detection) 알고리즘은 처음에 컴퓨터 네트워크 로그 마이닝을 위해 제안되었으나, 다차원적으로 동시에 발생하는 별개의 시계열 데이터에 적용하도록 일반화되었다. 이 알고리즘은 수많은 데이터 스트림에서 발생하는 값 사이의 강한 종속성를 찾는다. 찾아진 종속성은 와일드 카드를 허용하는 다차원적 이벤트 셋인 prefix와 suffix으로 구성된 <PS>형태이다.

3.3.2 WINEPI 알고리즘[15]

1995년도에 특별히 알고리즘에 대한 이름이 붙여지지 않고 제안[14]되었다가 1997년도에 상세히 정리하고 WINEPI란 알고리즘이 붙어서 제안되었다. 다른 알고리즘과 달리 원격 통신 네트워크 경보 로그로부터 빈발 시퀀스 발견을 위해 고안된 알고리즘이다. 여기서 사용하는 용어인 에피소드(episode)는 함께 발생하는 이벤트의 순서화된 부분적인 집합이다. 이 에피소드는 DAG(directed acyclic graphs)로 이루어진다. 또한 이벤트의 시간적 순서를 요구하는 직렬 시퀀스와 순서에 대한 제약이 없는 병렬 시퀀스에 모두 적용 가능하다는 것이 장점이다.

3.3.3 EVE 알고리즘[17]

EVE(Event distribution) 여러 프로세서 사이에서 요구되는

계산 작업과 메모리 사용을 적절하게 분배하는 병렬화에 사용하는 알고리즘이다. GSP 알고리즘을 확장한 이 알고리즘은 3개로 구성되어있다. 먼저 EVE-S(Simple EVE)로 전체 시퀀스를 각 프로세서에 평등하게 분배하며 많은 수의 객체와 짧은 시퀀스 데이터에 적합하다. EVE-R(EVE with Partial Replication)은 객체의 수가 프로세서의 수보다 더 작을 때, 전체 이벤트를 각 프로세서에 할당하기 위해 각 시퀀스를 분할하는 방법이다. 마지막으로 EVE-C(EVE for Complex Scenario)는 가장 복잡하고 긴 span-window 크기를 가지며 작은 수의 긴 시퀀스의 경우에 사용하는 알고리즘이다.

## 4. 응용 도메인 및 기술적인 문제 해결을 위한 연구

이상에서 언급한 것처럼 순차 패턴 마이닝은 시퀀스 데이터를 대상으로 하고 있다. 이런 데이터의 측면에서 보면 기존의 거래 데이터에서 텍스트, WWW, 네트워크, 환자의 이력, 기상 정보, 멀티미디어에서 DNA 시퀀스까지 거의 대부분의 데이터가 대상이 된다. 또한 이런 다양한 분야에서 순차 패턴 알고리즘을 각 분야에 적용하기 위한 많은 기술들이 연구 및 개발되고 있다. 예를 들어, 데이터가 계속 추가되는 incremental 데이터베이스에서 빈발 시퀀스를 찾기 위한 연구가 있었고 지금도 연구되고 있다.

또한 해시(hash) 구조와 같은 적절한 자료 구조를 선택에 대한 연구, 다차원 속성에 대한 처리를 위한 연구, 샘플링(sampling)이나 분해(partitioning)와 같은 기법에 대한 연구들이 이루어졌다.

## 5. 향후 순차 패턴 마이닝의 연구 방향

이 연구를 통해 많은 논문들이 대량의 데이터베이스에서 빈발 패턴을 좀더 효율적으로 찾기 위해 많은 노력들이 이루어 졌음을 알 수 있다. 이를 위해 후보 집합의 생성 비용과 탐색 공간을 줄이고 스캔하는 횟수를 줄이려는 연구를 하였다. 몇몇 연구에서는 사용자가 원하는 패턴을 찾기 위한 연구나 긴 패턴을 찾기 위한 연구가 이루어졌다. 데이터베이스의 크기는 지금보다 훨씬 커질 것이고 그 대상도 객체 지향 데이터베이스, 능동 데이터베이스, 공간 데이터베이스, 시간 데이터베이스, 시공간 데이터베이스[19]와 멀티미디어 데이터베이스 등과 같이 다양화 될 것이다.

데이터의 크기를 줄이고 마이닝 속도를 향상시키기 위한 연구는 앞으로도 계속 진행되어야 한다. 또한 SPIRIT알고리즘의 연장선에서 다양한 제약 조건을 통해 사용자가 원하는 데이터를 제공해야 한다. PSP 알고리즘이나 FreeSpan처럼 효율적인 데이터 구조를 적용하여 효율적인 탐사도 필요하다.

병렬 처리를 위해 순차 패턴을 이용한 EVE 알고리즘처럼 분산 처리에도 순차 패턴을 이용한 새로운 알고리즘이 연구 될 수 있다. 현재의 규칙 표현을 확장하여 사용자가 이해하기 쉽도록 하고 다양한 표현을 위한 연구도 필요하다.

네트워크의 발달과 인터넷의 확산으로 전세계가 하나로 연결된 상황에서 이런 연구를 통해 실시간/온라인 순차 패턴 마이닝을 위한 연구는 더욱 필요하다. 또 여기서는 여러 속성을 가지고 순차 패턴을 할 수 있도록 확장된 기법도 연구되어야 한다.

마지막으로 찾아진 빈발 패턴을 평가할 수 있는 평가 도구에 대한 연구가 필요하다.

## 6. 요약

이 논문에서는 순차 패턴에 관한 많은 알고리즘을 분류하고 분석하였다. 이 연구를 통해 실제 응용에서 순차 패턴 마이닝을 이용할 때, 적합한 알고리즘을 선택할 수 있게 하고 향후 새로운 순차 패턴 알고리즘을 연구할 수 있는 기반을 제시하였다.

그러나 순차 패턴 알고리즘의 대상이 되는 응용 도메인에 대한 체계적인 분류와 각 분류에 적합한 연구들에 대한 비교 및 분석이 필요하다. 또한 제시한 순차 패턴의 연구 방향에 따른 연구도 추가적으로 필요하다.

## 참고문헌


[1] R. Agrawal and R. Srikant, "Mining Sequential Patters," Proceedings of the 11th International Conference on Data Engineering, 1995.

[2] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," International Conference on Extending Database Technology, 1996.

[3] Roberto J. Bayardo Jr., "Efficiently Mining Long Patterns from Databases," Proceedings of ACM SIGMOD International Conference on Management of Data, 1998.

[4] F. Masseglia, F. Cathala and P. Poncelet, "The PSP approach for Mining Sequential Patterns," Proceedings of European Symposium on Principle of Data Mining and Knowledge Discovery, 1998.

[5] M. Garofalakis, R. Rastogi and K. Shim, "SPIRIT: Sequential Pattern Mining with Regular Expression Constraints," Proceeding of VLDB Journal, 1999.

[6] M. Garofalakis, R. Rastogi and Kyuseok Shim., "Mining Sequential Patterns with Regular Expression Constraints," IEEE Transactions on Knowledge and Data Engineering, 2002.

[7] Mohammed J. Zaki, "Efficient Enumeration of Frequent Sequences," International Conference on Information and Knowledge Management, 1998.

[8] Mohammed J. Zaki, "SPADE: An Efficient Algorithm for Mining Frequent Sequences," Machine Learning Journal, 2001.

[9] Mohammed J. Zaki, "Sequence Mining in Categorical Domains: Incorporating Constraints," Proceedings of the 9th international Conference on Information and Knowledge Management, 2000.

[10] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal and M. Hsu, "FreeSpan: Frequent Pattern-projected Sequential Pattern Mining," Proceedings of International Conference Knowledge Discovery and Data Mining, 2000.

[11] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal and M. Hsu, "PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth," Proceedings of International Conference Data Engineering, 2001.

[12] M. Lin, S. Lee and S. Wang, "DELISP: Efficient Discovery of Generalized Sequential Patterns by Delimited Pattern-Growth Technolog," PAKDD 2002.

[13] T. Oates, D. Gregory and Paul R. Cohen, "Detecting Complex Dependencies in Categorical Data," In Preliminary Papers of the 5th International Workshop on Artificial Intelligence and Statistics, 1995.

[14] H. Mannila, H. Toivonen and A. Inkeri Verkamo, "Discovering frequent episodes in sequences," In First International Conference on Knowledge Discovery and Data Mining, 1995.

[15] H. Mannila, H. Toivonen and A. I. Verkamo, "Discovering Frequent Episodes in Event Sequences," Report C-1997-15, University of Helsinki, Department of Computer Science, 1997.

[16] Mohammed J. Zaki, "Parallel Sequence Mining on SMP Machines," Workshop on Large-Scale Parallel KDD Systems (in conjunction 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining), 1999.

[17] Mahesh V. Joshi, G. Karypis and V. Kumar, "Parallel Algorithms for Mining Sequential Associations: Issues and Challenges," Technical Report No. 00-002, Department of Computer Science, University of Minnesota, 2000.

[18] Demiriz, "A. webSPADE: A Parallel Sequence Mining Algorithm to Analyze Web Log Data," In Proceedings of ICDM 2002.

[19] 이준욱, 백옥현, 류근호, "위치 기반 서비스를 위한 이동 객체의 시간 패턴 탐사 기법," 한국정보과학회 논문지, 제 29 권 5 호 2002.