데이타 축소와 군집화를 사용하는 시공간 데이타의 이산화 기법

(Discretizing Spatio-Temporal Data using Data Reduction and Clustering)

강 주 영 † 용 환 승 ††
(Juyoung Kang) (Hwan-Seung Yong)

요 약 항목 기반의 순차 패턴 마이닝 기법들을 시공간 데이타에 적용하기 위해서는 시공간 속성 값에 대한 적절한 이산화가 필수적이다. 본 논문에서는 입력 데이타의 시공간적 상관 정보를 유지함과 동시에 데이타 수를 축소시킴으로써 마이닝 프로세스의 효율성을 높이는 이산화 기법을 제안한다. 제안된 기법은 선 단순화를 사용하여 궤적에 대한 근사치를 구함으로써 마이닝 단계에서 처리할 데이타 크기를 축소시킨다. 또한 단순화 된 궤적을 유사한 시공간적 특성을 가지는 논리적 그룹으로 군집화하여 데이타의 분포를 고려한 이산화를 수행한다. 실험을 통해 제안된 기법이 마이닝 프로세스의 효율성을 높일 뿐 아니라 보다 직관적이고 해석이 용이한 패턴을 도출하는 것을 보였다.

키워드 : 시공간 데이타 마이닝, 이산화, 선 단순화, 군집화

Abstract To increase the efficiency of mining process and derive accurate spatio-temporal patterns, continuous values of attributes should be discretized prior to mining process. In this paper, we propose a discretization method which improves the mining efficiency by reducing the data size without losing the correlations in the data. The proposed method first abstracts original trajectories into approximations using line simplification and then groups them into similar clusters. Our experiments show that the proposed approach improves the mining efficiency as well as extracts more intuitive patterns compared to existing discretization methods.

Key words : Spatio-temporal data mining, Discretization, Line simplification, Clustering

1. 서 론

이동 통신 및 측위 기술의 발전으로 휴대 전화나 PDA와 같은 다양한 시스템으로부터 방대한 양의 이동 객체 데이타가 수집되고 있다. 이를 기반으로 이동 객체의 시공간적 패턴을 분석하여 고품질 위치 기반 서비스를 제공하고자 하는 연구가 증가하는 추세이다. 시공간 패턴 탐사 문제는 이동 객체의 위치 데이타 중 빈번하게 나타나는 순차 패턴을 찾는 과정으로 볼 수 있다.

순차 패턴 마이닝은 처음 Agrawal[1]에 의해 소개 된 이후 여러 연구를 통해 개선 및 확장 되어 왔다. 하지만 기존의 기법들은 항목 기반 트랜잭션 데이타베이스를 대상으로 하기 때문에 연속적인 속성 값을 가진 시공간 데이타에 이를 단순하게 적용할 수 없다. 또한 복잡한 연산을 필요로 하는 시공간 데이타의 특성상 적절한 시간 내에 활용 가능한 결과를 도출하는 것은 매우 어려운 문제이다. 따라서 시공간 패턴 마이닝을 위해서는 연속 값을 이산화 하여 순차 패턴 마이닝에 적용 가능하도록 변환하는 전처리 단계가 필수적이다. 시공간 데이타를 이산화하는 가장 전형적인 방법은 데이타 공간의 각 차원을 사용자가 정의한 n개의 구간으로 나누는 균일 격자(regular grid) 방법으로, 등간격 구간화 EQW(EQual interval Width)의 2차원적 확장으로 볼 수 있다. 이 방법은 단순하고 직관적이지만 입력 데이타 속성의 분포를 고려하지 않고 고정된 셀에 데이타를 할당하기 때문에 이산화 과정 동안 데이타 내의 시공간적 상관 정보를 잃을 수 있으며 따라서 마이닝 품질 또한 떨어질 수밖에 없다[2].

본 논문에서는 시공간 데이타의 이산화 문제를 제시하고 입력 데이타의 시공간적 특성을 고려하여 이산화를 수행하는 기법을 제안하였다. 제안된 기법은 선 단순화(line simplification)를 통해 이동 객체의 궤적 데이타를 단순화 한 후 이를 군집화 하여 전체 데이타 공간을 분할하는 영역들을 찾아낸다. 이러한 영역들은 데이타의 공간적 지역성과 방향 속성에 따른 군집화 결과이므로 이산화 후에도 원본 데이타의 시공간적 상관 정보를 유지하게 된다. 제안된 기법은 데이타를 압축된 형태로 표현함으로써 마이닝 단계에서 처리할 데이타의 수를 축소시키고 마이닝 성능을 향상시킬 수 있다.

· 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2006-511-D00311)
· 이 논문은 2008 한국컴퓨터종합학술대회에서 '데이타 축소와 군집화를 사용하는 시공간 데이타의 이산화 기법'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 이화여자대학교 컴퓨터학과
jykang@ewhain.net
†† 종신회원 : 이화여자대학교 컴퓨터학과 교수
hsyong@ewha.ac.kr

논문접수 : 2008년 8월 28일
심사완료 : 2008년 11월 16일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제1호(2009.1)

2. 관련 연구

연속 속성 값의 이산화 과정은 데이타 마이닝의 중요한 전처리 단계 중 하나로 데이타를 알고리즘이 요구하는 입력 형태로 변환시키며 학습 속도와 정확도를 향상시킨다. Hussain[3] 등은 이산화 기법들에 대한 실증적인 연구를 통해 학습의 결과에 미치는 영향을 연구하고 기존의 기법들을 광역/지역, 교사/비교사 등의 기준에 따라 분류 및 평가하는 계층 프레임워크를 제시하였다.

시공간 데이타를 이산화 하는 가장 일반적인 방법은 데이타 공간을 미리 정의된 고정 크기의 구간으로 분할하는 균일 격자 방법이다. [2]와 [4]에서는 사용자가 미리 설정한 공간 영역 중 하나에 데이타를 할당함으로써 이산화를 수행하였다. 이 방법은 공간 분할이 전문가에 의해 미리 설정되어야 한다는 단점을 가진다. 시공간 마이닝을 통해 유의미한 패턴을 찾기 위해서는 이산화 후에도 원본 데이타 내에 존재하는 패턴 정보를 유지해야 하며, 이산화 기준이 입력 데이타의 특성을 반영해야 한다. 최근 이동 객체의 이력 데이타로부터 주기 패턴을 탐사하기 제안된 기법에서는 밀도 기반 군집화를 기반으로 분할된 공간 영역을 얻어내었다[5]. 입력 데이타를 고려하여 이산화 한다는 점에서 본 논문의 기법과 유사하지만, 데이타를 단순히 변환할 뿐 데이타 수를 축소하지는 않는다. 또한 [6]은 분할 영역을 찾기 위해 반복적으로 영역들을 병합 및 재분할하기 위해 복잡한 연산을 반복적으로 수행하기 때문에 길이가 긴 이동 궤적에 대해서는 효율적인 성능을 보장하지 못한다.

3. 시공간 데이타의 이산화

이동 객체의 궤적은 시간에 따라 연속적으로 움직이는 객체의 공간적 위치 이력 데이타이다. 객체의 위치는 특정 시각(timestamp)에서의 위치를 나타내는 2차원 좌표 값으로 표현된다. 따라서 시공간 시퀀스 S는 다음과 같은 연속적인 위치 측정값의 집합으로 나타낼 수 있다.

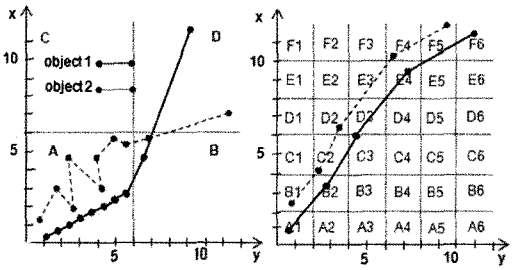
$$S = \{(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_n, y_n, t_n)\}$$

실제로 실측 위치 데이타는 노이즈를 포함하며, 같은 위치의 객체라 할지라도 움직임의 편차로 인해 서로 다른 측정값을 가질 수 있다. 따라서 이러한 데이타를 기반으로 순차 패턴 마이닝을 수행하기 위해서는 마이닝 단계 이전에 이러한 연속 속성 값을 이산화 해야 한다. 이를 위해 일반적으로 t_i시각의 위치 좌표 (x_i,y_i)를 이 위치가 해당되는 특정 공간 영역의 식별자로 변환한다. 연속된 측정 시각 사이의 시간 간격은 고정적이므로 시퀀스 S는 "$C_1C_2\cdots C_n$"와 같이 영역 식별자의 시퀀스 형태로 이산화 된다. 이 경우 객체의 이동 패턴 탐사 문제는 시퀀스들로부터 미리 정의된 최소 지지도 이상의 빈발 순차 패턴을 추출하는 작업으로 생각할 수 있다.

데이타의 시공간 연속 값을 이산화 하는 일반적인 방법은 데이타 공간에 대해 x, y 각 축을 사용자가 정의한 계수 C_x와 C_y를 이용하여 고정된 수의 구간으로 분할하고 데이타를 $C_x \times C_y$개의 격자 셀 중 공간적으로 사상 되는 셀 식별자 값으로 변환하는 방법이다. 따라서 연속적인 위치 값은 2차원 셀 식별자의 시퀀스 $\{C_1C_2...C_n\}$로 변형 된다. 이 방법은 단순하고 직관적인 반면, 데이타 공간을 각 차원에 독립적인 구간으로 분할하므로 이산화 과정 동안 객체의 움직임 정보를 잃을 수 있다. 그림 1(a)와 같이 셀이 너무 큰 경우 두 개의 서로 상이한 궤적이 동일한 시퀀스로 이산화 되게 된다. 반대로 셀의 크기가 지나치게 작은 경우, 그림 1(b)와 같이 매우 유사한 두 궤적이 서로 다른 시퀀스로 이산화 된다. 따라서 입력 데이타 내의 의미 있는 패턴들을 잃지 않기 위해서는 객체의 이동 변화에 대한 정보를 유지하며 이산화를 수행하도록 해야 한다.

4. 데이타 축소와 군집화 기반의 시공간 데이타 이산화 기법

본 논문에서는 입력 데이타의 시공간적 의미를 유지함과 동시에 데이타 수를 축소시켜 마이닝 프로세스의 효율성을 높이는 시공간 데이타 이산화 기법인 STEM (Spatio-TEmporal discretization of Moving objects trajectories)을 제안하였다. STEM은 우선 선 단순화를 통해 원본에 대한 근사 궤적을 구한 후, 이를 다차원 특성 벡터 형태로 변환한다. 마지막으로 군집화를 통해 데이타의 시공간적 변화에 따라 공간 분할하는 지역적 그룹들을 찾아낸다. 결과적으로 원본 데이타는 그 점이 속하는 군집 식별자의 연속적인 시퀀스로 이산화 된다.



(a) 간격이 넓은 격자에서의 이산화 (b) 간격이 좁은 격자에서의 이산화

grid (a)의 이산화 결과	grid (b)의 이산화 결과
S_1 = {A,A,A,A,A,A,A,A,B,D}	S_1 = {A1,B2,C3,E4,F6}
S_2 = {A,A,A,A,A,A,A,A,A,B,D}	S_2 = {B1,C2,D2,F4,F5}

(c) 각 격자에 의한 이산화 결과

그림 1 균일격자를 이용한 이동 객체 궤적의 이산화

4.1 선단순화를 이용한 궤적 근사

EQW와 같은 단순 이산화 기법은 원본 데이타 수만큼의 이산화 된 데이타를 유지하기 때문에 장기간의 궤적 데이타는 대단히 긴 시퀀스로 이산화 된다. 이러한 시퀀스를 대상으로 마이닝을 수행하는 경우 연산양은 폭발적으로 증가하게 된다. 따라서 시공간 데이타를 위한 효율적인 이산화 기법은 원본 데이타 크기를 축소시킬 수 있어야 한다. 선 단순화는 결정론적 오차 범위 안에서 선을 압축하는 방법이다. STEM에서는 여러 방법들 중 수학적 우수성이 검증된 DP(*Douglas-Peucker*) 알고리즘을 사용하여 선 단순화를 수행한다[7]. DP 알고리즘은 궤적 T:{$p_1,p_2,\cdots,p_n$} 점집합을 부분집합 T':{$p'_1,p'_2,\cdots,p'_s$}$\subseteq T$, $s\leq n$으로 재귀적으로 분해한다. 즉, 사용자가 정의한 임계값 ε_l(tolerance)에 대해 원본 p_{i-1}과 p_i 사이의 모든 점을 수직 거리 최대 ε_l인 선분 $p'_{i-1}p'_i$으로 단순화 한다. 기본 DP 알고리즘은 최악의 경우 시간 복잡도 $\Theta(n^2)$를 가지지만, 데이타 수가 적절한 경우 O($nlogn$)의 성능을 보장한다고 알려져 있다.

4.2 특성 벡터 도출과 정규화

이전 단계에서 원본 궤적은 단순화된 유향선분들로 추상화된다. 이를 기반으로 이산화를 위한 공간 분할을 찾아내기 위해서 이 선분들을 다차원 특성 벡터의 형태로 변환한다. 두 객체의 위치가 같은 영역에 사상되는 경우라도 이동 방향에 따라 움직임의 의미가 전혀 다를 수 있기 때문에 객체의 이동 방향을 나타내는 선분 간 상대 각도 값을 특성 벡터 내에 포함시킴으로써 시공간 객체의 움직임을 더욱 정확하게 표현하도록 한다. p_l에서 p_r을 두 끝점으로 하는 선분의 특성 벡터 v는 선분의 두 끝점과 선분과 x축이 이루는 각도 Θ로 이루어진다(v={p_l,p_r,Θ}). 결과적으로 STEM은 전 단계에서 얻은 근사 궤적 T:{$p'_1,\cdots,p'_m$}로부터 V:{$v_1,\cdots,v_k$} ($1<k<m-1$)를 도출한다(v_k={p_{k-1},p_k,Θ_k}). 서로 다른 범위를 가지는 특성 벡터의 값을 단순히 비교하는 것은 의미가 없기 때문에 최대-최소 정규화를 통해 벡터 값의 범위를 [0,1] 사이 값으로 선형 변환함으로써 벡터 값의 영향을 평준화한다. 정규화 된 특성 벡터 값 v'는 $v'=((v-min)/(max-min))\times(newmax-newmin)+newmin$)으로, $newmin$과 $newmax$는 각각 0과 1로 설정하였다.

4.3 특성 벡터의 군집화

객체의 변화는 데이타 공간 전체에 균일하게 나타나는 것이 아니기 때문에 입력 데이타 분포를 기반으로 공간을 적절히 분할하기 위해서는 이전 단계에서 얻은 특성 벡터를 유사한 특징을 가지는 집합으로 군집화 하는 단계가 필요하다. 일반적으로 군집화 알고리즘은 이차 시간 복잡도를 가지는 것으로 알려져 있으며, 이는 이산화 프로세스에 성능 병목 현상을 일으킬 수 있다. 따라서 군집화 기법은 적절한 성능을 보장하기 위해서 계산 복잡도를 줄이고, 벡터들을 시공간적 근접성에 따라 그룹화하기 위한 임계값 설정이 가능해야 한다. 본 논문에서는 BIRCH[8]의 초기 군집화(pre-clustering) 단계를 적용하였다. 이 군집화 기법은 입력 크기에 대해 선형 시간 복잡도를 가지며 데이타의 요약 정보를 CF-트리라는 간결한 트리 구조에 저장한다. 트리의 단말노드는 사용자가 지정한 근접도 임계값 ε_c 내에 존재하는 점들로 이루어진 하나의 군집을 나타낸다. 데이타를 점진적으로 읽어가면서 CF-트리를 탐색하여 가장 가까운 단말노드를 찾고, 만약 입력 데이타가 그 단말노드가 나타내는 군집의 임계값 내부에 위치한다면 해당 노드에 삽입한다. 그렇지 않은 경우 단말노드를 분할하고, 노드 내 점들을 분할된 노드로 재분배 한다(그림 2).

STEM의 군집화 알고리즘은 임계값 ε_c에 대해 다차원 특성 벡터 값으로부터 유사한 특성을 가지는 선분 그룹 들을 찾아내고 이를 기반으로 데이타 공간을 분할한다. 최종적으로 원본 궤적은 궤적 내의 각 점이 해당되는 군집 식별자의 일련의 시퀀스로 이산화 된다.

5. 실 험

이 절에서는 합성 데이타를 기반으로 제안된 기법의 마이닝 및 공간 효율성을 비교 평가하고, 마이닝 결과를 가시화 하여 이산화 기법이 마이닝 프로세스에 미치는 영향을 분석한다. 본 실험은 PentiumD 3.4GHz, WindowsXP 환경에서 실행되었다. 구현을 위해 기하함수 C++ Library[1]와 BIRCH 코드[2]를 참고하였다.

```
Procedure Clustering (V, εc)
// V : The set of vectors,  εc : Clustering threshold
begin
1: for each vector v ∈ V do {
2:   if (v.clusterID = null){
3:       find the closest leaf node in CF-tree from v
4:       find the closest entry in the node from v
5:       if (v is not within the threshold distance of the entry)
6 :          if ( the node is full ) {
7 :               modifying the related node /*splitting */
8:                modifying path to the node }
9 :          else
10 :              add a new entry to the node
11:          update CF-tree
12:          assign clusterID to v
13:      else
14:          assign clusterID to v
15:  }
16: }
End
```

그림 2 특성 벡터의 군집화 알고리즘

1) http://www.geometryalgorithms.com/
2) http://pages.cs.wisc.edu/~vganti/birchcode/

표 1 합성 데이타의 매개변수 설정

Dataset	Parameters
DS_1, DS_2	TS= 100/200, NS= 5 NO= 100, D= normal
DS_3~DS_7	TS= 200, NS= 5, NO= 50~300, D= normal
DS_8	TS= 80, NS= 1, NO = 5, D= normal

5.1 합성 데이타 집합

본 실험에서 시공간 데이타를 생성하기 위해 사용한 합성 데이타 생성기는 G-TERD[9] 이다. 궤적은 객체가 하루 동안의 이동 이력 데이타로 가정하였으며 데이타 생성에 사용된 매개 변수 설정은 표 1과 같다. TS는 궤적 내 timeslot의 수, NS는 이동 패턴을 구분하는 시나리오의 수, NO는 시나리오 별 객체 수, D는 객체들의 중심 분포를 나타낸다. 데이타는 1000×1000 크기(unit)의 작업 공간 내에 분포 하며 객체의 움직임을 조절하기 위한 상세 설정은 [9]에서 제안한 바를 따랐다.

5.2 마이닝 결과 비교

이산화 기법이 마이닝에 미치는 영향을 평가하기 위해 *PrefixSpan*[10] 순차 패턴 마이닝을 수행하였다. *PrefixSpan* 알고리즘은 Han 등에 의해 제안된 패턴 증식 접근법 기반의 순차 패턴 마이닝 알고리즘[3]이다. 그림 3(a)와 3(b)의 회색 음영 부분은 데이타 집합 DS_8의 원본 궤적을 나타낸다. 10×10 격자를 사용한 EQW 기반 결과를 그림 3(a)에, STEM 기반의 마이닝 결과를 그림 3(b)에 가시화 하였다(선 단순화 임계값 ε_l=0.1, 군집 임계값 ε_c=0.3). 그림 3(b)는 STEM이 x축 범위 [70,90]과 y축 [50,70] 사이의 경우와 같이 데이타의 이동 패턴에 변화가 발생한 점을 기준으로 영역을 분할하는 (S_6,S_7) 것을 보여준다. 이산화 기법에 따른 마이닝 결과를 비교하기 위해 최대 빈발 패턴을 비교하였다. 최대 빈발 패턴은 빈발한 초패턴(super pattern)이 존재하지 않는 패턴으로 최대 빈발 패턴의 모든 부분 집합은 빈발하다. EQW의 최대 빈발 패턴은 {E_{25},E_{36},E_{47},E_{58},E_{69}, E_{68},E_{78},E_{76}} (반복되는 동일 식별자 생략)로, 실제는 동일 셀 식별자가 반복적으로 나타나는 길이 33의 패턴이다. 반면 STEM은 {S_1,S_2,S_3,S_4,S_5,S_6,S_7,S_8}와 같이 간결하고 직관적인 패턴을 생성하였다. 즉, STEM이 EQW에 비해 객체의 이동에 대해 더욱 해석하기 쉽고 직관적인 패턴을 생성한다는 것을 알 수 있다.

5.3 공간 효율성과 마이닝 효율성

선 단순화를 통한 데이타 축소 효과에 따른 공간 효율성을 평가하기 위해 데이타 축소율을 측정하였다. 데이타 축소율은 단순화된 궤적에 포함된 점의 총 수를 원본 궤적의 점의 수로 나누어서 얻을 수 있다. 이 때,

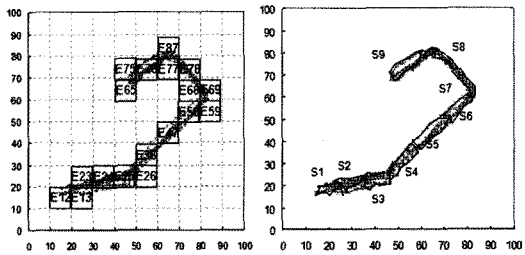


그림 3 EQW와 STEM의 이산화 결과 비교

실제 저장 공간은 (1-데이타 축소율) 만큼 절약 된다고 예측할 수 있다. 그림 4에서 볼 수 있듯이 임계값 ε_l을 작게 설정한 경우 즉, 원본 궤적과 단순화된 궤적과의 오차 값이 작은 경우에도 데이타 크기를 상당히 줄여 저장 공간의 효율성을 높이는 것을 알 수 있다.

다음 실험으로 이산화가 마이닝 성능에 미치는 영향을 평가하기 위해 이산화와 마이닝 알고리즘의 수행 시간을 합한 전체 수행 시간을 측정하였다. EQW의 격자 개수는 총 50×50개로 설정하고. 데이타 집합 D_3~D_7에 대해 선 단순화 임계값 ε_l을 0.2와 0.5로, 군집화 임계값 ε_c을 0.1로 설정하고 *PrefixSpan*을 실행하였다. 표 2는 최소 지지도가 5%일 때 수행 시간을 초 단위로 나타낸 것이다. EQW는 각 차원을 고정 크기의 구간으로 단순하게 분할하기 때문에 이산화 과정 자체에서는 빠른 성능을 보인다. STEM은 선 단순화와 군집화를 처리하기 때문에 최악의 경우 $O(n^2)$의 시간 복잡도를 가지는 반면 전체 수행 시간의 측면에서는 마이닝 단계의 효율성이 이산화 수행 속도를 보완하기 때문에 EQW보다 좋은 성능을 보이는 것을 알 수 있다. 또한 EQW는 데이타 축소 효과가 없고, 이산화 된 셀의 수가 많기 때문에 데이타 수 증가에 따라 성능이 현저하게 떨어지게 된다.

5.4 원본 데이타에 대한 근사도

이산화는 원본 데이타에 대한 근사화를 수행하는 과

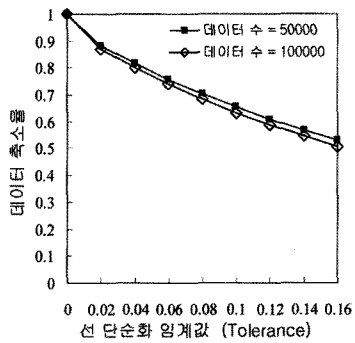


그림 4 임계값에 따른 STEM의 데이타 축소율

3) http://illimine.cs.uiuc.edu/download/

표 2 STEM과 EQW의 전체 수행 시간(이산화+마이닝)

데이타 수	STEM(ε_l = 0.2)		STEM(ε_l = 0.5)		EQW	
	Disc.	Overall	Disc.	Overall	Disc.	Overall
50000	0.579	1.297	0.359	0.391	0.015	40.015
100000	1.156	1.735	0.703	0.781	0.047	16.594
150000	1.641	1.922	1.078	1.140	0.062	4.8900
200000	2.172	2.047	1.438	1.532	0.093	8.0780
250000	2.719	3.031	1.828	1.937	0.109	30.327
300000	3.328	3.329	2.110	2.251	0.141	16.453

정으로 볼 수 있으며, 따라서 마이닝 결과와 별도로 이산화 결과와 원본 데이타의 차이를 기반으로 이산화 기법을 평가할 수 있다. 즉, 원본 데이타와 이산화 결과 사이의 근사 오차의 총 합이 적을수록 이산화 성능이 우수하다고 판단 가능하다. EQW 근사 오차는 원본 데이타 점과 그 점이 사상된 셀의 중앙점 사이 거리로 나타낼 수 있다. 반면 STEM의 경우는 이산화 과정에서 데이타 축소가 일어나기 때문에 단순화 된 선분으로부터 임계값 ε_l 이내의 모든 점들이 선분의 양 끝점으로 나타내어진다. 따라서 STEM의 근사 오차를 구하기 위해 원본 점들과 그 점이 단순화 된 선분 위로 사상되는 점 사이의 거리를 근사 오차로 정의하였다. 표 3은 표 2와 동일한 실험 조건에서 근사 오차의 총합을 비교하여 나타낸 것이다. STEM의 경우 원본 점과 단순화 된 점 사이 거리는 임계값을 넘지 않기 때문에 EQW에 비해 근사 오차가 작다. EQW의 오차를 줄이기 위해 분할 단위를 줄일 수 있으나 이 경우 전체 셀의 수가 증가하여 마이닝 성능이 저하되고 추출된 패턴의 직관적 이해도 또한 떨어질 것으로 추측할 수 있다.

6. 결 론

본 논문에서는 시공간 데이타를 효율적으로 이산화하기 위한 이산화 기법을 제시하였다. 제안된 기법은 선 단순화를 이용해 원본 궤적에 대한 근사값을 구하고 이를 군집화 함으로써 이산화 과정 동안 데이타의 시공간적 상관 정보를 유지한다. 제안된 기법은 기존의 기법들과는 달리 원본 데이타의 시공간 속성 정보를 고려하여 분할 영역을 생성하므로 마이닝 단계에서 더 직관적이고 이해하기 쉬운 패턴을 생성한다. 또한 데이타의 수를 축소시켜 마이닝의 수행 복잡도를 줄일 수 있다. 실험을 통해 제안된 기법이 전체 마이닝 프로세스의 효율성을 높이고 추출된 패턴의 해석성을 높이는 것을 보였다.

표 3 원본 데이타에 대한 근사오차의 총합

데이타 수	STEM(ε_l = 0.2)	STEM(ε_l = 0.5)	EQW
50000	2076.7	6695.8	382233.9
100000	4138.6	13497.8	763469.7
150000	6281.1	20325.5	1146332.0
200000	8289.0	27065.4	1529809.9
250000	10312.9	33747.0	1915558.3
300000	12347.7	40693.6	2297077.5

참 고 문 헌

[1] Agrawal R. and Srikant R., Mining Sequential Patterns, In Proc. of ICDE, pp. 3-14, Mar., 1995.

[2] Tsoukatos, I. and Gunopulos, D., *Efficient mining of spatiotemporal patterns,* In Proc. of Int'l. Symp. on Spatial and Temporal Databases., pp. 425-442, Jul., 2001.

[3] Hussain, F., Liu, H., Tan, C. L. and M. Dash., *Discretization: An Enabling Technique.* Journal of Data Mining and Knowledge Discovery, Vol.6, No.4, pp. 393-423, Jun., 2002.

[4] Yavas, G., Katsaros, D., Ulusoy, O. and Manolopoulos. Y., *A data mining approach for location prediction in mobile environments*, Data and Knowledge Engineering. Vol.54, No.2, pp. 121-146, Aug., 2005,

[5] Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y. and Cheung, D. W., *Mining, indexing, and querying historical spatiotemporal data,* In Proc. of 10th Int'l Conference on KDD, pp. 236-245, Aug., 2004.

[6] Cao, H., Mamoulis, N., Cheung, D. W., *Mining frequent spatio-temporal sequential patterns.*, In Proc. of Data Mining, pp. 82-89, Nov., 2005.

[7] Hershberger,J. and Snoeyink,J., *Speeding up the Douglas-Peucker line-simplification algorithm*, In Proc.of 5th h Int'l Symp. on Data Handling, pp. 134-143, Aug., 1992.

[8] Zhang, T., Ramakrishnan, R. and Livny. M., *BIRCH: An efficient data clustering method for very large databases.* In Proc. of SIGMOD, pp. 103-114, Jun., 1996.

[9] Tzouramanis, T., Vassilakopoulos, M. and Manolopoulos. Y., *On the generation of time-evolving regional data*, Geoinformatica, Vol.6, No.3, pp. 207-231, Sep., 2002.

[10] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. and Hsu. M.C., *PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth*, In Proc. of 17th Int'l Conference on Data Engineering, pp. 215-224, Apr., 2001.