시퀀스 데이터베이스를 위한 타임 워핑을 지원하는 효과적인 서브시퀀스 매칭

박 상 현*, 김 상 욱**, 조 준 서*

IBM T.J. Watson 연구소*
강원대학교 컴퓨터정보통신공학부**

Effective Subsequence Matching Supporting Time Warping in Sequence Databases

Sang-Hyun Park*, Sang-Wook Kim**, June-Suh Cho*

IBM T.J. Watson Research Center*
Division of Computer, Information, and Communications Engineering
Kangwon National University**

요약문

본 논문에서는 대용량 시퀀스 데이터베이스에서 타임 워핑을 지원하는 인덱스 기반 서브시퀀스 매칭에 관하여 논의한다. 타임 워핑은 시퀀스의 길이가 서로 다른 경우에도 유사한 패턴을 갖는 시퀀스들을 찾을 수 있도록 해 준다. 최근의 연구에서 타임 워핑을 지원하는 효과적인 전체 매칭 기법이 제안된 바 있다. 본 연구에서는 이 기존의 연구에 슬라이딩 윈도우 개념을 결합하는 새로운 기법을 제안한다. 인덱싱을 위하여, 각 슬라이딩 윈도우와 대응되는 서브시퀀스로부터 특징 벡터를 추출하고, 이 특징 벡터를 인덱싱 애트리뷰트로 사용하는 다차원 인덱스를 구성한다. 질의 처리를 위하여, 조건을 만족하는 질의 접두어들에 대한 특징 벡터들을 이용하여 인덱스 검색을 수행한다. 제안된 기법은 대용량의 데이터베이스에서도 효과적인 서브시퀀스 매칭을 지원한다. 본 연구에서는 제안된 기법이 착오 기각을 유발시키지 않음을 증명하고, 실험을 통하여 제안된 기법의 우수성을 규명한다.

1. 서론

시퀀스 데이터베이스(sequence database)란 객체의 변화되는 값들의 연속으로 구성된 데이터 시퀀스(data sequence)들의 집합이다[Agr93]. 본 논문에서는 시퀀스 데이터베이스에서 타임 워핑을 지원하는 효과적인 서브시퀀스 매칭(subsequence matching)에 관하여 논의한다.

서브시퀀스 매칭은 질의 시퀀스와 유사한 서브시퀀스를 포함하는 시퀀스를 검색하는 연산이다. 타임 워핑(time-warping)은 시퀀스내의 각 요소 값을 임의의 수만큼 반복시키는 것을 허용하는 변환이다[Yi98]. 예를 들어, 타임 워핑에 의하여 두 시퀀스 S = <20, 21, 21, 20, 20, 23, 23, 23>와 Q = <20, 20, 21, 20, 23>를 동일한 시퀀스 <20, 20, 21, 21, 20, 20, 23, 23, 23>으로 변환시킬 수 있다. 타임 워핑 후의 두 시퀀스들 간의 거리를 타임 워핑 거리(time warping distance)라 정의한다. 타임 워핑은 데이터베이스내의 시퀀스들의 길이가 서로 달라서 유클리드 거리를 이용하여 유사 정도를 직접 측정할 수 없는 경우에 매우 유용하다.

참고 문헌 [Kim01]에서는 인덱스를 기반으로 타임 워핑을 지원하는 효과적인 전체 매칭(whole matching) 기법을 제안한 바 있다. 이 기법은 데이터베이스 내의 각 시퀀스로부터 추출된 특징(feature)들에 대하여 다차원 인덱스를 구축하고, 특징 공간(feature space)에서 사용하기 위한 새로운 거리 함수 D_{tw_lb}를 사용한다. 이 거리 함수 D_{tw_lb}는 삼각형 부등식 성질을 만족하는 타임 워핑 거리의 하한 함수(lower bound function)이다. 이 결과, 이 기법은 착오 기각 없이 고속의 유사 검색을 지원할 수 있다. 본 논문에서는 이 기법을 LB_Filter라 부른다. 본 논문에서는 LB_Filter를 확장함으로써 타임 워핑을 지원하는 *서브시퀀스 매칭*을 착오 기각 없이 효과적으로 처리하는 기법을 제안한다.

LB_Filter를 응용한 가장 손쉬운 방법은 데이터베이스 내에 존재하는 *모든 가능한* 서브시퀀스들에 대하여 특징을 추출하고, 이 특징들에 대하여 다차원 인덱스를 구성하는 것이다. 또한, 서브시퀀스 매칭 시에는 D_{tw_lb}를 이용한 인덱스 검색을 통하여 후보들을 파악하고, 이들에 대한 착오 채택(false alarm) 여부를 조사한다. 이 방법은 LB_Filter를 손쉽게 적용할 수 있다는 장점이 있다. 그러나 길이가 L인 하나의 시퀀스 내에서 추출 가능한 모든 서브시퀀스의 수는 (L+1)*L/2로서 매우 많다. 따라서 이 방식은 저장 공간 오버헤드 측면에서 큰 무리가 있다.

이러한 문제를 해결하기 위하여 본 논문에서는 슬라이딩 윈도우(sliding window)[Fal94] 개념을 이용한 *접두어 질의 기법(prefix-querying method)*을 제안한다. 즉, 데이터베이스 내의 각 시퀀스를 고정된 길이의 슬라이딩 윈도우들로 분할하고, 각 윈도우로부터 참고 문헌 [Kim01]의 방식으로 특징들을 추출한다. 또한, 이 특징들을 인덱싱 애트리뷰트로 이용하여 전체 데이터베이스를 위한 다차원 인덱스를 구성한다. 서브시퀀스 매칭 시에는 질의 시퀀스의 가능한 접두어에 대하여 특징들을 추출하고, 이 특징들과 허용치 ε을 가지고 다차원 인덱스를 검색한다. 인덱스 검색 결과로 반환된 후보들에 대하여 착오 채택 여부를 검사함으로써 최종 질의 결과를 구한다. 제안된 기법의 견고성(robustness)를 규명하기 위하여 서브시퀀스 매칭에서 착오 기각이 발생되지 않음을 증명한다. 또한, 실험에 의한 성능 분석을 통하여 제안된 기법의 우수성을 제시한다.

2. 관련 연구

2.1. 용어 정의

시퀀스 S(= <s[1], s[2], ..., s[|S|]>)는 실수인 요소 값들의 연속이다. 여기서 |S|는 시퀀스의 길이이며, s[i]는 S의 i번째 요소를 의미한다. First(S)와 Last(S)는 각각 S의 첫 번째 요소 s[1]과 마지막 요소 s[|S|]를 의미한다. Rest(S)는 s[1]을 제외한 S의 나머지 요소들로 구성되는 시퀀스 <s[2], s[3], ..., s[|S|]>를 의미한다. S[i:j]는 S의 서브시퀀스로서 i번째 요소에서부터 j번째 요소까지를 포함한다. <>은 요소가 존재하지 않는 널 시퀀스(null sequence)를 의미한다.

길이 n을 갖는 두 시퀀스 S와 Q의 유사한 정도를 측정하기 위하여 다음과 같은 거리 함수 L_p가 널리 사용된다. L_1은 맨하탄 거리(Manhattan distance), L_2는 유클리드 거리(Euclidean distance), L_∞은 대응되는 각 쌍의 거리 중 최대 거리를 의미한다. 거리 함수 L_p는 대상이 되는 두 시퀀스의 길이가 같아야 한다는 제한이 있다.

$$L_p(S,Q) = \left(\sum_{i=1}^{n} |\, s[i] - q[i] \,|^p\right)^{1/p}, \quad 1 \le p \le \infty.$$

두 시퀀스 S와 Q간의 타임 워핑 변환을 기반으로 한 타임 워핑 거리(time warping distance) D_{tw}는 다음과 같이 재귀적으로 정의된다:

정의 1:

(1) $D_{tw}(<>, <>) = 0$,
(2) $D_{tw}(S, <>) = D_{tw}(<>, Q) = \infty$,
(3) $D_{tw}(S, Q) = D_{base}(First(S), First(Q)) +$
$MIN(D_{tw}(S,Rest(Q)),D_{tw}(Rest(S),Q),D_{tw}(Rest(S),Rest(Q)))$

□

여기서, min은 인자들 중 가장 작은 값을 가지는 것을 취하는 함수이며, D_{base}는 기본 거리 함수로서 L_p 중 응용에 적합한 임의의 것을 선택하여 사용할 수 있다. 특히 본 연구에서는 요소 반복을 통하여 변환된 두 시퀀스간의 거리 함수 D_{base}로서 L_∞를 사용한다. D_{base}로서 L_∞를 사용하는 주된 이유는 사용자의 질의 작성의 부담을 덜도록 하기 위해서이다[Kim01][1].

2.2 LB_Filter

참고 문헌 [Kim01]에서는 전체 매칭을 위한 인덱스 기반 유사 검색 기법을 제안하였다. 이 연구에서는 삼각형 부등식을 만족하는 D_{tw}의 새로운 하한 거리 함수 $D_{tw\text{-}lb}$를 고안하였다. $D_{tw\text{-}lb}$는 각 시퀀스 S로부터 타임 워핑과 무관한 네 개의 특징 <First(S), Last(S), Greatest(S), Smallest(S)>를 함수의 인자로서 사용한다. 여기서 First(S), Last(S), Greatest(S), Smallest(S)는 각각 S의 첫 값, 마지막 값, 최대 값, 최소 값을 의미한다.

이러한 특징들을 인자로 사용하는 하한 함수 $D_{tw\text{-}lb}$는 다음과 같이 정의된다.

정의 2:

$$D_{tw\text{-}lb}(S, Q) = L_p(Feature(S), Feature(Q))$$

여기서, Feature(S) = <First(S), Last(S), Greatest(S), Smallest(S)>, Feature(Q) = <First(Q), Last(Q), Greatest(Q), Smallest(Q)>이며, p는 주어진 응용에서 사용하는 D_{tw}가 채택하는 p와 동일하게 결정된다.

□

이 기법에서는 이러한 네 특징들을 인덱싱 애트리뷰트로 사용하는 사차원 인덱스를 구성하고, 이 인덱스와 $D_{tw\text{-}lb}$를 이용한 필터링 단계를 통하여 전체 매칭을 매우 효율적으로 처리한다. 본 논문에서는 이 기법을 LB_Filter라 부른다.

LB_Filter를 서브시퀀스 매칭에 응용하기 위한 가장 단순한 방법은 데이터베이스 내에 존재한 *모든 가능한* 서브시퀀스들에 대하여 네 개의 특징들을 추출하고, 이들에 대하여 다차원 인덱스를 구성하는 것이다. 서브시퀀스 매칭 시에는 이 인덱스를 이용하여 후보들을 효과적으로 파악할 수 있다. 그러나 길이가 L인 시퀀스 내에 존재하는 가능한 서브시퀀스의 수는 (L+1)*L/2로서 매우 많다. 예를 들어, 데이터베이스 내에 평균 길이 1,000인 1,000개의 시퀀스들이 저장되어 있는 경우 500,500,000개의 특징들을 포함하는 사차원 인덱스를 구성해야 한다. 이 결과, 다차원 인덱스 내에 저장해야 하는 특징들의 수는 시퀀스 길이의 이차 함수로 증가하고, 시퀀스 수의 일차 함수로 증가하는 형태를 가진다. 따라서 대형 데이터베이스 환경에서는 저장 공간 오버헤드 측면에서 큰 무리가 있다.

3. 제안하는 기법

3.1. 기본 전략

본 논문에서는 위와 같은 LB_Filter의 단순한 적용으로 인한 문제점을 해결함으로써 대형 데이터베이스에서도 무리 없이 동작할 수 있는 효과적인 서브시퀀스 매칭 기법을 제안한다. 제안하는 기법은 착오 기각 배제를 위한 이론적 배경으로서 다음과 같은 정리를 기반으로 한다.

정리 1:

임의의 두 시퀀스 s, q, 그리고 임의의 양수 w ($1 \le w \le |s|$)에 대하여, 만일 s와 q의 타임 워핑 거리가 ε 이내이면, s의 접두어 s[1:w]와의 타임 워핑 거리가 ε 이내인 q의 접두어(prefix)가 반드시 존재한다. 즉, 아래의 공식이 성립한다.

$$(\exists x)\ D_{tw}(s, q) < \varepsilon \ \Rightarrow\ (D_{tw}(s[1:w], q[1:x]) < \varepsilon)$$

증명: 생략

□

LB_Filter에서 사용하였던 거리 함수 $D_{tw\text{-}lb}$는 타임 워핑 거리 D_{tw}의 하한 함수이므로, 정리 1로부터 다음과 같은 따름 정리 1을 쉽게 유도할 수 있다.

따름 정리 1:

임의의 두 시퀀스 s, q, 그리고 임의의 양수 w ($1 \le w \le |s|$)에 대하여, 다음의 식이 항상 성립한다.

$$(\exists x)\ D_{tw}(s, q) \le \varepsilon \ \Rightarrow\ (D_{tw\text{-}lb}(s[1:w], q[1:x]) \le \varepsilon),$$ 여기서 $1 \le j \le |q|$.

□

따름 정리 1을 효과적인 서브시퀀스 매칭에 활용하기 위하여 본 논문에서는 슬라이딩 윈도우(sliding window)[Fal94] 개념을 이용한다. 즉, 데이터베이스 내의 각 시퀀스를 길이가 w인 슬라이딩 윈도우들로 분할하고, 각 윈도우로부터 네 개의 특징 First(w), Last(w), Greatest(w), Smallest(w)를 추출한다. 이 네 개의 특징들을 인덱싱 애트리뷰트로 이용하여 전체 데이터베이스를 위한 사차원 인덱스를 구성한다. 길이가 L인 시퀀스로부터 추출되는 길이가 w인 슬라이딩 윈도우의 수는 (L-w+1) 개이다. 이것은 모든 가능한 서브시퀀스의 수 (L+1)*L/2와 비교하여 훨씬 작은 수이므로 인덱스의 저장 공간 측면에서 실용 가능하다.

윈도우의 길이 w는 서브시퀀스 매칭의 처리 성능과 매우 밀접한 관계가 있다. 크기 w인 슬라이딩 윈도우들에 대하여 인덱스가 구성되어 있다고 가정하자. 서브시퀀스 s, 질의 시퀀스 q에 대하여, w의 크기가 작을수록 $D_{tw}(s, q)$와 $D_{tw\text{-}lb}(s[1:w], q[1:x])$ ($x \le |q|$)의 차이는 커지며, 이 결과, $D_{tw\text{-}lb}$를 이용한 인덱스 필터링 단계에서 반환되는 착오 채택의 수는 많아진다. 또한, q와 w의 차이가 커질수록 s[1:w]와 비교해야 하는 q의 접두어의 개수가 많아진다. 이 두 가지 사실은 윈도우의 길이가 질의 시퀀스 길이 보다 작을수록 서브시퀀스 매칭의 처리 성능이 저하됨을 의미하는 것이다. 그 반면, 윈도우의 길이가 질의 시퀀스의 길이보다 큰 경우에는 인덱스를 이용한 필터링이 불가능하다.

본 논문에서는 이러한 두 가지 문제를 동시에 해결하기 위한 합리적인 대안으로서 minQLen과 maxWarpRatio[Ber96]의 두 가지 개념을 사용한다. minQLen은 응용에서 사용될 수 있는 최소의 질의 시퀀스 길이를 의미하며, maxWarpRatio는 타임 워핑 변환에 의하여 하나의 시퀀스 요소가 반복될 수 있는 최대 수를 의미한다[Ber96]. 본 연구에서는 응용의 특성에 따라 minQLen과 maxWarpRatio가 미리 결정된다는 가정 하에, 윈도우의 크기를 $w = \left\lceil \frac{\min QLen}{\max WarpRatio} \right\rceil$로 결정한다.

즉, w는 최소 길이의 질의 시퀀스와 매치될 수 있는 데이터베이스 내의 최소 서브시퀀스의 길이를 의미하는 것이다.

이러한 크기 w의 윈도우들을 대상으로 구성된 인덱스를 이용하여 질의 시퀀스 q를 위한 서브시퀀스 매칭은 다음과 같이 처리된다. 먼저, 인덱스 내에 저장된 크기 w인 서브시퀀스(즉, 윈도우)와 매치될 수 있는 q의 접두어의 최소 및 최대 길이를 결정한다. 이러한 접두어의 최소 길이 minPrefixLen은 $\left\lceil \frac{w}{\max WarpRatio} \right\rceil$이며, 최대 길이 maxPrefixLen은 w*maxWarpRatio(= minQLen)이다. 다음으로, q의 각 접두어 q[1:j] (minPrefixLen≤j≤maxPrefixLen)에 대하여, 네 개의 특징

1) L_∞의 사용은 사용자의 질의 작성 부담을 덜기 위한 유사 모델상의 단순한 변화를 의미한다. 따라서 D_{base}로서 L_1혹은 L_2를 사용하는 경우에도 본 논문에서 제안하는 인덱스 기반 서브시퀀스 매칭 기법은 그대로 사용할 수 있음을 밝혀두고자 한다.

First(q[1:j]), Last(q[1:j]), Greatest(q[1:j]), Smallest(q[1:j])를 추출하고, 이 특징들과 D_{tw_lb} 그리고 ε을 이용하여 인덱스 검색을 수행한다. 끝으로, 인덱스 검색을 통하여 반환된 후보들을 실제로 액세스하여 착오 제거를 제거한다.

3.2. 추가 성능 개선 방안

제안된 질의 처리 알고리즘에서는 질의 시퀀스 q의 각 접두어 q[1:j] (minPrefixLen≤j≤maxPrefixLen)에 대하여 개별적으로 인덱스 검색을 수행한다. 따라서 접두어의 수가 커지는 경우, 인덱스 검색의 비용이 커지므로 서브시퀀스 매칭의 처리 성능이 저하될 수 있다. 본 논문에서는 각각의 접두어에 대하여 개별적으로 인덱스 검색을 수행하지 않고, 그룹으로 병합한 다수의 접두어들에 대하여 한번의 인덱스 검색을 수행하는 방법을 사용한다.

질의 영역을 위한 MBR 구성 방법을 요약하면 다음과 같다.

(1) 하나의 접두어와 일대일 대응되는 다수의 질의 MBR들을 구성하는 방법

(2) 모든 접두어들을 포함하는 단 하나의 질의 MBR을 구성하는 방법

(3) 접두어들을 다수의 그룹들로 분할하고, 각 그룹과 대응되는 MBR을 다수 구성하는 방법

일반적으로 접두어의 수가 많아지는 경우에는 MBR의 지나친 크기 증가를 억제하기 위하여 다수의 MBR로 구성하는 것이 효과적이다. 이러한 MBR 구성에 관한 사항은 본 연구의 주요 논점이 아니므로, 본 논문에서는 전체 문제를 단순화하기 위하여 방법 (1)과 방법 (2)만을 대상으로 하고, 구체적인 MBR 구성 방안에 관해서는 향후 연구로 다루고자 한다. 본 논문에서는 방법 (1)과 (2)를 채택한 질의 처리 알고리즘을 각각 TW_Subseq_Matching1과 TW_Subseq_Matching2라 부른다.

4. 성능 분석

본 장에서는 성능 평가를 위하여 제안된 TW_Subseq_Matching1와 TW_Subseq_Matching2의 성능을 기존의 LB_Scan[Yi98], ST_Filter[Par00]의 성능과 비교 검토한다[2]. 본 연구에서 사용한 데이터 S&P_Data는 미국의 S&P 500 실제 주식 데이터이며, 평균 길이가 231인 545개의 시퀀스들로 구성된다.

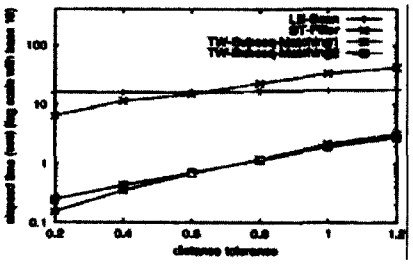

그림 5.1. 허용치 ε의 변화에 따르는 평균 수행 시간의 비교(S&P_Data).

동일한 상황에서의 성능 평가를 위하여 본 실험에서는 D_{tw}를 위한 기본 거리 함수 D_{base}로서 L_1를 사용하는 기존의 기법도 본 논문에서 채택하는 L_∞을 사용하도록 한다. 또한, ST_Filter를 위한 도메인 분류 방법으로서 최대 엔트로피 기법(maximum entropy method)을 이용하여 ST_Filter가 50개의 구간을 갖도록 한다. 제안된 기법을 위한 다차원 인덱스로는 현재 가장 널리 채택되고 있는 R*-트리를 사용하여 실험한다. 사용된 페이지 크기는 1KB이다. 허용치 ε를 0.2, 0.4, 0.6, 0.8로 변화하면서 각 기법의 성능을 비교하였다[3]. 사용된 다른 인자로서 maxWarpRatio, minQLen, avgQLen을 5, 50, 90으로 각각 설정하였다. 여기서 aveQLen은 평균 질의 시퀀스의 길이를 나타낸다.

그림 5.1은 S&P_Data에 대한 각 기법의 실험 결과를 나타낸 것이다. 실험 결과에 의하면 제안된 기법은 기존의 기법과 비교하여 허용치에 따라 6배에서 42배까지의 성능 개선 효과를 가지는 것으로 나타났다.

5. 결론

본 논문에서는 기존의 LB_Filter를 확장함으로써 타임 워핑을 지원하는 *서브시퀀스 매칭*을 착오 기각 없이 효과적으로 처리하는 *슬라이딩 윈도우 기반 접두어 질의 기법(prefix-querying method based on sliding windows)*을 제안하였다.

제안된 기법에서는 슬라이딩 윈도우(sliding window)[Fal94] 개념을 이용한 인덱싱 전략을 사용한다. 제안된 기법의 견고성(robustness)을 증명하기 위하여 제안된 기법을 이용한 서브시퀀스 매칭에서 착오 기각이 발생되지 않음을 증명하였다. 또한, 실제 데이터를 이용한 실험을 통하여 제안된 기법의 우수성을 규명하였다.

6. 감사의 글

본 논문은 한국학술진흥재단 선도연구자 연구비 지원에 의하여 연구되었습니다. (과제번호: KRF-2000-041-E00258)

7. 참고 문헌

[Agr93] R. Agrawal, C. Faloutsos, and A. Swami, "Efficient Similarity Search in Sequence Databases," In Proc. Int'l. Conference on Foundations of Data Organization and Algorithms, FODO, pp. 69-84, Oct. 1993.

[Ber96] D. J. Berndt and J. Clifford, "Finding Patterns in Time Series: A Dynamic Programming Approach," Advances in Knowledge Discovery and Data Mining, pp. 229-248, 1996.

[Fal94] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast Subsequence Matching in Time-series Databases," In Proc. Int'l. Conf. on Management of Data, ACM SIGMOD, pp. 419-429, May 1994.

[Kim01] S. W. Kim, S. H. Park, and W. W. Chu, "An Index-Based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases," In Proc. Intl. Conf. on Data Engineering, IEEE, pp. 607-614, 2001.

[Par00] S. H. Park et al., "Efficient Searches for Similar Subsequences of Difference Lengths in Sequence Databases," In Proc. Int'l. Conf. on Data Engineering, IEEE, pp. 23-32, 2000.

[Par01] S. H. Park, S. W. Kim, and W. W. Chu, "Segment-Based Approach for Subsequence Searches in Sequence Databases", In Proc. ACM Intl. Symp. on Applied Computing (ACM SAC), pp. 248-252, 2001.

[Yi98] B. K. Yi, H. V. Jagadish, and C. Faloutos, "Efficient Retrieval of Similar Time Sequences Under Time Warping," In Proc. Int'l. Conf. on Data Engineering, IEEE, pp. 201-208, 1998.

2) 다양한 실험을 통하여 제안된 기법과 기존 기법의 성능상의 여러 가지 특성을 관찰할 수 있었다. 그러나 본 논문에서는 지면 관계상 그림 5.1에 나타난 허용치에 따르는 성능상의 경향만을 제시한다.

초기에는 참고 문헌 [Yi98]과 [Par01]에 나타난 FastMap 기법과 SBASS 기법도 모두 성능 비교 대상으로 고려하였다. 그러나, FastMap 기법은 착오 기각을 유발하며, SBASS 기법은 유사 모델 자체가 기존의 기법들과 크게 다르다. 따라서 이러한 직접적인 성능상의 비교는 큰 의미가 없는 것으로 판단되어 최종 실험에서는 제외하였다.

3) 허용치가 0.2인 경우는 평균 8개의 최종 결과를 반환하였고, 허용치가 1.2인 경우는 평균 1703개의 최종 결과를 반환하였다.