시퀀스 데이터베이스를 위한 타임 워핑 기반 유사 검색

김 상 욱*, 박 상 현**

강원대학교 컴퓨터정보통신공학부*

IBM T.J. Watson 연구소**

Time-Warping-Based Similarity Search in Sequence Databases

Sang-Wook Kim*, Sang-Hyun Park**

Division of Computer, Information, and Communications Engineering
Kangwon National University*

IBM T.J. Watson Research Center**

요약문

본 논문에서는 대형 시퀀스 데이터베이스에서 타임 워핑을 지원하는 유사 검색을 효과적으로 처리하는 방안에 관하여 논의한다. 타임 워핑은 시퀀스의 길이가 서로 다른 경우에도 유사한 패턴을 갖는 시퀀스들을 찾을 수 있도록 해 주는 변환이다. 타임 워핑 거리는 삼각형 부등식 성질을 만족하지 못하므로 기존의 기법들은 착오 기각 없이 다차원 인덱스를 사용할 수 없었다. 본 논문에서는 타임 워핑을 지원하는 새로운 인덱스 기반 유사 검색 기법을 제안한다. 제안된 기법의 주요 목표는 착오 기각 없이 대형 데이터베이스에서도 좋은 검색 성능을 보장하는 것이다. 다양한 실험을 통하여 제안된 기법의 우수성을 규명한다. 실험 결과에 의하면, 제안된 기법은 기존의 기법과 비교하여 약 4배에서 43배까지의 성능 개선 효과를 가지는 것으로 나타났다.

1. 서론

시퀀스 데이터베이스(sequence database)란 객체의 변화되는 값들의 연속으로 구성된 데이터 시퀀스들의 집합이다[1]. 대표적인 예로는 주가 데이터, 환율 데이터, 기온 데이터, 제품 판매량 데이터, 기업 성장률 데이터 등이 있다. 유사 검색(similarity search)이란 주어진 질의 시퀀스(query sequence)와 변화의 패턴이 유사한 시퀀스들을 시퀀스 데이터베이스로부터 찾아내는 연산이다[1]. 이러한 유사 검색은 데이터 마이닝(data mining) 및 데이터 웨어하우징(data warehousing) 분야에서 중요한 연산으로 사용된다.

유사 검색에 관한 기존의 많은 연구에서는 길이 n의 시퀀스를 n 차원 공간상의 한 점으로 간주하고, 두 시퀀스들간의 유사한 정도를 측정하기 위하여 두 점들간의 유클리드 거리(Euclidean distance)를 이용한다[1]. 유클리드 거리만을 이용한 유사 검색을 통해서는 사용자가 원하는 시퀀스들을 검색하지 못하는 경우가 빈번하게 발생한다. 따라서 응용 분야에 적합한 유사 모델(similarity model)을 적절하게 정의할 수 있도록 변환(transform)을 지원하기도 한다.

타임 워핑(time warping)[3][5][6]은 시퀀스내의 각 요소 값을 임의의 수만큼 반복시키는 것을 허용하는 변환이다. 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.

기존의 연구에서는 효율적인 유사 검색을 위하여 다차원 인덱스(multidimensional index)[2]를 사용한다[1]. 대부분의 인덱스들은 채택하는 거리 함수가 삼각형 부등식 성질(triangle inequality)[7]을 만족한다는 것을 전제로 한다. 만일, 이 성질을 만족하지 못하는 거리 함수를 이용하는 경우에는 유사 검색 시 착오 기각(false dismissal)이 발생된다[6]. 착오 기각이란 실제 질의 결과로 반환되어야 할 질의 시퀀스와 유사한 시퀀스를 올바르게 찾아내지 못하는 현상이다[1]. 참고 문헌 [6]에서는 타임 워핑 거리가 삼각형 부등식 성질을 만족하지 못함을 증명하고, 착오 기각을 허용하지 않는 응용에서 타임 워핑을 지원하는 유사 검색을 처리할 때에는 거리 함수 기반 인덱스를 사용할 수 없다고 주장한 바 있다.

참고 문헌 [3]와 [6]에서는 인덱스 없이 시퀀스들을 모두 액세스함으로써 타임 워핑 지원 유사 검색을 처리하는 방법을 제안하였다. 그러나 대규모의 데이터베이스 환경에서는 이와 같이 인덱스를 사용하지 않는 경우, 검색 성능이 심각하게 저하된다. 참고 문헌 [6]에서는 또한 FastMap[4]을 이용하여 k(<<n) 차원 공간내의 점들로 변환된 시퀀스들을 대상으로 다차원 인덱스를 구성함으로써 검색 성능을 개선하는 방식을 제안하였다. 그러나 이 방식은 착오 기각을 유발시킨다는 심각한 문제점을 가지므로, 이를 허용하는 제한된 응용에 한해서만 사용될 수 있다. 참고 문헌 [5]에서는 거리 함수를 기반으로 하지 않는 서픽스 트리(suffix tree)를 사용함으로써 착오 기각을 허용하지 않으면서 부분 매칭 시의 검색 성능을 개선시킬 수 있는 방식을 제안하였다. 그러나 이 방식은 좋은 성능을 보장하는 분류 작업(categorization)이 매우 복잡하며, 또한 전체 매칭 시에는 트리의 크기가 매우 커지므로 검색 성능이 저하된다는 문제점을 갖는다.

본 논문에서는 타임 워핑을 지원하는 유사 검색을 처리하기 위한 효율적인 방법에 관하여 논의한다. 본 연구의 목표는 착오 기각 발생의 방지와 빠른 검색 성능을 동시에 보장하는 것이다. 본 연구에서는 새로운 거리 함수를 고안하고, 이 거리 함수를 기반으로 구성된 다차원 인덱스를 이용하여 타임 워핑을 지원하는 유사 검색을 빠르게 처리할 수 있는 새로운 기법을 제안한다. 제안된 기법의 견고성(robustness)을 규명하기 위하여 유사 검색에서 착오 기각이 발생되지 않음을 보인다. 또한, 다양한 실험에 의한 성능 분석을 통하여 제안된 기법의 우수성을 제시한다. 제안된 기법은 거리 함수를 이용하는 인덱스를 기반으로 하는 최초의 타임 워핑 지원 유사 검색 처리 기법이라는 점에서 큰 의미가 있다.

2. 유사 검색 모델

시퀀스 데이터베이스는 다양한 길이를 갖는 시퀀스들의 집합으로 구성된다. 시퀀스 $S(= \langle s_1, s_2, ..., s_{|S|}\rangle)$는 실수인 요소 값들의 연속이다. 여기서 $|S|$는 시퀀스의 길이이며, s_i는 S의 i번째 요소를 의미한다. First(S)와 Last(S)는 각각 S의 첫 번째 요소 s1과 마지막 요소 $s_{|S|}$를 의미한다. Rest(S)는 s_1을 제외한 S의 나머지 요소들로 구성되는 시퀀스 $\langle s_2, ..., s_{|S|}\rangle$를 의미한다. <>은 요소가 존재하지 않는 널 시퀀스(null sequence)를 의미한다. 데이터베이스 내에 저장된 시퀀스를 데이터 시퀀스라 하고, 유사 검색 질의에 주어지는 시퀀스를 질의 시퀀스라 한다.

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 D_{tw}는 다음과 같이 재귀적으로 정의된다[3]:

정의 1:

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

□

본 연구에서는 서로 다른 길이를 가지는 두 시퀀스들의 유사한 정도를 나타내는 척도로서 정의 1에 나타난 타임 워핑 거리 D_{tw}를 사용한다. 특히, 요소 반복을 통하여 변환된 두 시퀀스 간의 거리 함수 D_{base}로서 L_∞를 사용한다. D_{base}로서 L_1을 사용하는 참고 문헌 [3][5][6]과 달리 본 연구에서 L_∞를 사용하는 주된 이유는 사용자의 질의 작성의 부담을 덜도록 하기 위해서이다. 질의 시퀀스 Q와 허용치 ε이 주어지는 유사 검색 질의에서 $D_{tw}(S, Q)$의 값이 ε이하인 데이터 시퀀스 S들은 Q와 유사한 시퀀스로서 간주된다. 이는 S의 타임 워핑 변환된 시퀀스의 각 요소가 Q의 타임 워핑 변환된 시퀀스의 대응되는 요소의 일정 범위 ε내에 존재함을 의미한다.

3. 제안하는 기법

참고 문헌 [6]에서는 타임 워핑 거리가 삼각형 부등식 성질을 만족하지 못함을 보였으며, 착오 기각을 허용하지 않는 응용에서는 타임 워핑 지원 유사 검색의 처리를 위하여 인덱스를 사용할 수 없음을 주장한 바 있다. 그러나 대용량의 데이터베이스 환경에서 이와 같이 인덱스를 사용하지 않는 경우, 검색 성능이 심각하게 저하된다.

본 연구에서는 이에 대한 해결 방법으로서 삼각형 부등식 성질을 만족하는 타임 워핑 거리의 하한 함수(lower bound function)를 고안하고, 이를 기반으로 인덱스를 구성하는 전략을 사용한다.

하한 함수를 정의하기 위해서는 이 함수에서 인자로 사용될 시퀀스의 특징들을 먼저 추출해야 한다. 특징 추출이 어려운 이유는 타임 워핑 거리를 계산하기 위하여 같은 시퀀스가 질의 시퀀스에 따라 다양한 형태로 변환되기 때문이다. 즉, 요소 반복을 적용하는 위치나 횟수에는 특별한 제약이 없으므로, 비교되는 질의 시퀀스에 따라 같은 시퀀스라도 다양한 길이와 요소 값을 갖는 새로운 시퀀스로 변환될 수 있다. 그러나 시퀀스로부터 추출되는 특징은 인덱스 구성을 목적으로 하므로 질의 시퀀스와 독립적인 고유의 성질을 가져야 한다. 이것은 특징 추출이 시퀀스를 인자로 하는 함수(function)의 형태로 표현되어야 함을 의미한다.

본 연구에서는 하한 함수를 위한 인자로서 사용될 시퀀스 S의 특징들로서 첫 값인 First(S), 마지막 값인 Last(S), 요소들 중 최대 값인 Greatest(S), 요소들 중 최소 값인 Smallest(S)를 선정한다. 이들은 주어진 질의 시퀀스와의 타임 워핑 거리 계산을 위한 어떠한 형태의 요소 반복에도 변하지 않는 고정된 특징들이다. 시퀀스 S의 네 특징들로 구성되는 4-터플 레코드를 Feature(S)라 표기한다. 이러한 특징들을 인자로 사용하는 타임 워핑 거리 D_{tw}의 하한 함수 D_{tw_lb}는 다음과 같이 정의된다.

정의 2: $D_{tw\_lb}(S, Q) = L_\infty(Feature(S), Feature(Q))$
여기서 Feature(S) = <First(S), Last(S), Greatest(S), Smallest(S)>, Feature(Q) = <First(Q), Last(Q), Greatest(Q), Smallest(Q)> 이다. □

다음은 D_{tw_lb}의 성질을 나타내는 두 개의 정리이다.

정리 1:

임의의 두 시퀀스 $S = \langle s_1, s_2, ..., s_n\rangle$, $Q = \langle q_1, q_2, ..., q_m\rangle$와 임의의 값 ε에 대하여 다음이 항상 성립한다.

$$D_{tw}(S, Q) \le \varepsilon \Rightarrow D_{tw\_lb}(S, Q) \le \varepsilon$$

증명: 생략 □

정리 2:

임의의 세 시퀀스 X, Y, Z에 대하여 다음이 항상 성립한다.

$$D_{tw\_lb}(X, Z) \le D_{tw\_lb}(X, Y) + D_{tw\_lb}(Y, Z)$$

증명: 생략 □

정리 1은 유사 검색 질의를 처리할 때, D_{tw} 대신 D_{tw_lb}를 사용하는 경우에도 착오 기각이 발생하지 않음을 의미하는 것이다. 정리 2는 유사 검색 질의를 처리할 때, 삼각형 부등식 성질의 만족하는 D_{tw_lb}를 거리 함수로 하는 사차원 인덱스를 사용할 수 있음을 의미하는 것이다. 따라서 위의 두 정리들은 새로운 거리 함수 D_{tw_lb}를 기반으로 구성된 인덱스를 이용하여 타임 워핑 지원 유사 검색 질의를 착오 기각 없이 처리할 수 있음을 증명하는 것이다.

유사 검색 질의는 다음의 방식으로 처리된다. (1) 인덱스 구성: 데이터베이스 내의 각 시퀀스 S에 대하여 Feature(S)에 해당되는 네 값을 추출하여 이를 기반으로 사차원 인덱스를 구성한다. (2) 질의 처리: 질의 시퀀스 Q에 대하여 Feature(Q)에 해당되는 네 값을 추출하고, 사차원 인덱스를 이용한 정사각형 형태의 영역 질의를 수행한다. 이때, 추출된 네 값은 영역 질의의 중심점이 되며, ε은 질의의 범위가 된다. 거리 함수로는 D_{tw_lb}가 사용된다. 영역 검색의 결과로 반환된 후보들에 대하여 다시 D_{tw}를 계산하여 최종 결과를 구한다.

4. 성능 분석

본 연구에서는 평균 길이가 231인 545개의 시퀀스들로 구성된 미국의 S&P_Data를 이용한 실험을 통하여 제안된 기법의 성능을 분석한다. 각 데이터에 대하여 100개의 유사 검색 질의를 수행한 후, 나타난 평균 수행 시간(elapsed time)을 성능 평가 지수로 사용한다.

성능 평가는 다음의 네 가지 서로 다른 기법들을 대상으로 한다. Ours는 본 논문에서 제안된 기법으로서 시퀀스의 네 가지 특징들을 대상으로 구성된 다차원 인덱스 R*-트리[2]를 이용하는 방식이다. 또한, 성능 비교를 위한 기존의 기법으로서 Naive_Scan[3], LB_Scan[6], ST_Filter[5]의 세 가지를 사용한다

그림 1은 실험 결과를 나타낸 것이다. 가로축은 허용치 ε을 나타내며, 세로축은 실행 시간을 나타낸다. 실험 결과에 의하면, ST_Filter는 Naive_Scan보다도 떨어지는 성능을 가지는 것으로 나타났다. 그 근본적인 이유는 ST_Filter가 공통 심볼들이 많이 발생하는 서브시퀀스 환경을 대상으로 고안된 기법이기 때문이다. 따라서 ST_Filter는 서브시퀀스 매칭에는 유용하나, 전체 매칭에서는 적합하지 않다.

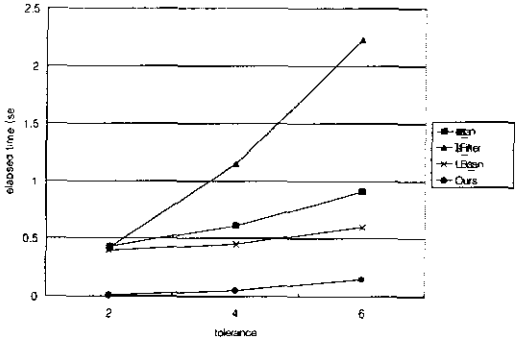

그림 1. S&P_Data를 이용한 실행 시간의 비교.

기존의 기법들 중에서는 LB_Scan이 가장 좋은 성능을 가지는 것으로 나타났다. 전체 시퀀스들을 액세스한다는 점에서는 LB_Scan과 Naive_Scan가 동일하지만, LB_Scan는 하한 함수를 사용함으로써 CPU 비용을 절감할 수 있기 때문이다. 제안된 기법은 LB_Scan에 비교하여 허용치에 따라 약 4배에서 43배까지 나은 성능을 보였다. 이것은 제안된 기법이 데이터의 4% 미만의 작은 R*-트리의 극히 일부분만을 탐색하며, 이 탐색에 의한 필터링 효과가 매우 뛰어남을 의미하는 것이다. 또한, 이러한 성능 개선 효과는 허용치가 작아질수록 더욱 두드러짐을 볼 수 있다. 실제 응용에서 요구하는 질의 결과의 수가 작다는 것을 고려할 때, 이러한 경향은 매우 바람직한 것이다.

5. 결론

본 논문에서는 착오 기각 발생의 방지와 빠른 검색 성능을 동시에 보장하는 새로운 타임 워핑 지원 유사 검색 처리 기법을 제안하였다. 제안된 기법은 먼저 새롭게 고안된 거리 함수 D_{tw_lb}를 이용하여 거리 함수 기반 다차원 인덱스를 구성하고, 이를 이용하여 타임 워핑 지원 유사 검색을 빠르게 처리한다. D_{tw_lb}가 D_{tw}의 하한 함수인 동시에 삼각형 부등식 성질을 만족한다는 것을 보임으로써 제안된 기법에서 착오 기각이 발생하지 않음을 증명하였다. 제안된 기법은 거리 함수를 기반으로 하는 최초의 인덱스 기반 타임 워핑 지원 유사 검색 기법이라는 점에서 큰 의미가 있다.

6. 감사의 글

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

7. 참고 문헌

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

[2] N. Beckmann et al., "The R*-tree: An Efficient and Robus Access Method for Points and Rectangles," In *Proc. Int' Conf. on Management of Data*, ACM SIGMOD, p 322-331, May 1990.

[3] D. J. Berndt and J. Clifford, "Finding Patterns in Time Series: A Dynamic Programming Approach," *Advances i Knowledge Discovery and Data Mining*, pp. 229-248, 199

[4] C. Faloutsos and K. I. Lin, "FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets," In *Proc. Int'l. Con on Management of Data*, ACM SIGMOD, pp. 163-17 1995.

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

[6] B. K. Yi, H. V. Jagadish, and C. Faloutos, "Efficien Retrieval ofSimilar Time Sequences Under Time Warping," In *Proc. Int'l. Conf. on Data Engineering*, IE pp. 201-208, 1998.

[7] F. P. Preparata and M. Shamos, *Computational Geomet An Introduction*, Springer-Verlag, 1985.