윈도우 제약 조건을 가지는 시간 왜곡 변환 기반 유사 시퀀스 검색

김인태°, 송병호+, 이석호
서울대학교 전기·컴퓨터공학부
+상명대학교 소프트웨어학부
kit123@db.snu.ac.kr, bhsong@sangmyung.ac.kr, shlee@cse.snu.ac.kr

Similar Sequence Searching under Time Warping with Window constraint

Intae Kim°, Byoungho Song+, Sukho Lee
School of Electrical Engineering and Computer Science, Seoul National University
+Division of Software Science, Sangmyung University

요 약

유사 시퀀스 검색에서 시간 왜곡 변환을 지원하기 위한 연구가 최근 활발히 이루어지고 있다. 음성 인식과 같은 몇몇 응용에서는 시간 왜곡 변환을 적용할 때 과도한 타이밍의 차이는 허용하지 않을 필요가 있다. 그래서 대부분의 경우 윈도우라는 제약 조건을 추가하게 된다. 이 논문에서는 윈도우 제약 조건이 있을 때 시간 왜곡 변환을 지원하는 유사 검색 방법으로 세그먼트 분할 기법(Segment Partition Approach: SPA)을 제안한다. SPA는 각 시퀀스를 세그먼트로 분할한 뒤 특징을 추출하여 다차원 인덱스를 구성한다. 유사 검색 질의를 수행할 때 이 인덱스를 검색하여 질의 시퀀스와 유사할 가능성이 큰 후보들을 빠르게 찾아낼 수 있고 찾아낸 후보들에 대해서만 정확한 시간 왜곡 변환 거리를 계산하기 때문에 전체 질의 처리 시간을 단축할 수 있다. SPA는 순차 검색에 비하여 좋은 성능을 보이며, 특히 거리 허용치가 작을 때 더욱 우수한 성능을 보인다.

1. 서 론

시퀀스는 시간의 흐름에 따라 순차적으로 생성되는 데이타의 모임이다. 데이타베이스에 저장된 시퀀스를 데이타 시퀀스(data sequence)라 부르며, 사용자에 의해 주어진 질의 시퀀스(query sequence)와 유사한 데이타 시퀀스를 검색하는 것을 유사 시퀀스 검색(similar sequence searching)이라고 한다[1].

유사 시퀀스 검색에 관한 기존의 많은 연구들은 유사성의 척도로써 유클리드 거리(euclidean distance)를 사용한다. 그러나, 응용 분야에 따라서 유클리드 거리만을 이용한 유사 검색으로는 사용자가 원하는 시퀀스들을 찾아내지 못하는 경우가 빈번하게 발생한다. 따라서 음성, 오디오, 의료정보 등을 위한 유사 검색에서는 시간 왜곡 변환 거리(time warping distance)[2]를 많이 사용하고 있다.

시간 왜곡 변환은 시퀀스내의 각 요소 값을 임의의 수만큼 반복시키는 것을 허용하는 변환이다. 예를 들어, 두 시퀀스 S=<10,12,12,11,14,14>와 Q=<10,10,12,11,11,14>는 시간 왜곡 변환을 적용하면 동일한 시퀀스 <10,10,12,12,11,11,14,14>로 변환시킬 수 있다.

그러나 시간 왜곡 변환은 한 시퀀스의 하나의 요소가 다른 시퀀스의 너무 많은 요소들과 대응될 수 있다. 예를 들어 두 시퀀스 S=<10,10,10,10,10,10,10,1>과 Q=<10,1,1,1,1,1,1,1>은 시간 왜곡 변환을 적용하면 동일한 시퀀스 <10,10,10,10,10,10,10,1,1,1,1,1,1,1>로 변환시킬 수 있지만 이 때 S의 마지막 원소 1과 Q의 첫 번째 원소 10은 각각 상대 시퀀스의 거의 대부분의 원소들과 대응되게 된다. 그런데 음성 인식과 같은 몇몇 응용에서는 이러한 시간축에 대한 과도한 가속 또는 감속을 허용하지 않아야 한다.

이러한 문제를 해결하기 위해서 윈도우 제약 조건을 이용할 수 있다[2, 3]. 시퀀스 S와 Q를 비교하는 경우 S의 i번째 원소와 Q의 j번째 원소가 시간 왜곡 변환을 통해 서로 대응될 때 i와 j가 주어진 윈도우 크기 w이내에 있어야 한다는 조건이다.

이 논문에서는 윈도우 제약 조건이 있을 때 시간 왜곡 변환 기반 유사 검색을 효율적으로 처리하는 방법을 제안한다. 제안하는 기법은 각 시퀀스를 세그먼트로 분할한 뒤 특징을 추출한다. 시간 왜곡 변환 거리의 하한 함수를 정의하고, 이 거리 함수를 기반으로 구성된 다차원 인덱스를 통하여 질의 시퀀스와 유사할 가능성이 많은 후보 시퀀스를 선택함으로써 실제 거리를 계산할 시퀀스의 수를 대폭 줄일 수 있으며, 따라서 착오기각(false dismissal)없이 전체 검색 시간을 크게 줄일 수 있다.

논문의 구성은 다음과 같다. 2장에서는 시간 왜곡 변환을 지원하는 유사 검색에 대한 기존의 연구를 살펴본다. 3장에서는 거리함수와 윈도우를 정의하고 4장에서는 세그먼트 분할 기법(Segment Partition Approach: SPA)에 대해 설명한다. 5장에서는 실제 데이타를 이용한 실험을 통해 성능을 평가하고 6장에서는 결론을 내린다.

2. 관련 연구

[2]에서는 인덱스 없이 모든 데이타 시퀀스들과 질의 시퀀스 간의 거리를 계산하여 시간 왜곡 변환을 지원하는 유사 시퀀스 검색을 제안하였다. 시간 왜곡 변환 거리를 계산하는 비용은 많은 CPU연산시간을 필요로 하기 때문에 이를 줄이기 위하여 [4]에서는 시간 왜곡 변환 거리의 하한 함수를 제안하였다. 그러나 대규모의 데이타베이스에서는 이와 같이 인덱스를 사용하지 않고 전체 데이타 시퀀스를 모두 읽어들여서 질의를 처리하는 경우 검색 성능이 심각하게 저하되는 문제가 있다.

[5]는 거리 함수를 이용하는 인덱스를 기반으로 한 최초의 시간 왜곡 변환 지원 유사 시퀀스 검색이다. 이 기법은 시간 왜곡 변환 거리의 하한 함수를 제안하고 이를 기반으로 인덱스를 구성하는 전략을 사용한다. 하한 함수를 위한 인자로써 각 시퀀스들로부터 4개의 값(시작, 끝, 최대, 최소)을 추출하여 그 시퀀스를 대표하는 값으로 정의한 다음 이를 기반으로 하한 함수를 정의하고 인덱스를 구성하여 질의와 유사할 가능성이 있는 후보 시퀀스를 착오기각 없이 찾아낼 수 있다.

[6]에서는 서픽스 트리를 이용하여 부분 매칭시의 성능을 개선하는 방안을 제시하였다. 그러나 이 방식은 전체 매칭시에는 트리의 크기가 매우 커지므로 검색 성능이 저하된다는 문제점이 있다.

3. 유사 검색 모델

시퀀스 데이타베이스는 다양한 길이의 시퀀스들로 구성된다. 시퀀스 $S(=<s_1,s_2,...,s_{|S|}>)$는 실수인 요소 값들의 연속이다. 여기서 $|S|$는 시퀀스의 길이이며, s_i는 S의 i번째 요소를 의미한다. Rest(S)는 s_1을 제외한 S의 나머지 요소들로 구성되는 시퀀스이다. <>은 요소가 없는 널 시퀀스(null sequence)를 의미한다.

길이가 n인 두 시퀀스 S와 Q의 거리를 계산하는 함수로 다음과 같은 거리함수 L_p가 널리 사용된다.

$$L_p(S,Q)=\left(\sum_{i=1}^{n}|s_i-q_i|^p\right)^{\frac{1}{p}}$$

L_1은 맨하탄 거리(Manhattan distance), L_2는 유클리드 거리(Euclidean distance), L_∞는 대응되는 각 쌍의 거리 중에서 최대값을 의미한다.

이 연구에서는 시간 왜곡 변환을 행한 뒤 두 시퀀스를 비교하는 거리 함수로 L_∞를 사용한다. 따라서 두 시퀀스 S와 Q간의 시간 왜곡 변환 거리 D_{tw}는 다음과 같이 정의된다[5].

$$D_{tw}(\langle\rangle,\langle\rangle)=0$$
$$D_{tw}(S,\langle\rangle)=D_{tw}(\langle\rangle,Q)=\infty$$

$$D_{tw}(S,Q)=\max\begin{cases}|s_1-q_1|\\ \min\begin{cases}D_{tw}(S,Rest(Q))\\ D_{tw}(Rest(S),Q)\\ D_{tw}(Rest(S),Rest(Q))\end{cases}\end{cases}$$

시간 왜곡 변환시의 윈도우 제약 조건은 다음과 같이 정의된다. 윈도우 크기가 w로 주어졌을 경우에 시간 왜곡 변환을 통하여 시퀀스 S의 j번째 요소와 Q의 j번째 요소가 대응될 때 i와 j는 다음의 식을 만족하여야 한다[2, 3].

$$\left|i-\frac{|S|}{|Q|}j\right|\le w$$

4. 세그먼트 분할 기법(SPA)

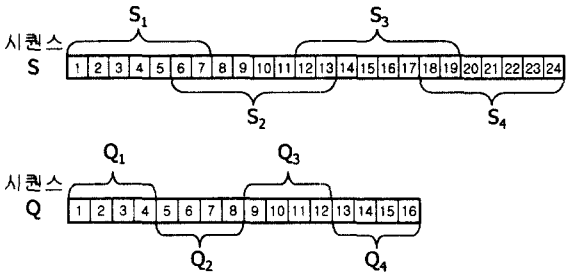
이 논문에서는 시간 왜곡 변환 거리의 하한 함수를 정의하고 이를 기반으로 인덱스를 구성하는 전략을 사용한다.

하한 함수를 정의하기 위해서는 인자로 사용할 시퀀스들의 특징을 추출하여야 한다. 그러나 시간 왜곡 변환을 허용할 경우 같은 시퀀스라 할지라도 질의 시퀀스에 따라 다양한 형태로 변형될 수 있기 때문에 특징 추출이 매우 어렵다. 그러나 이 연구에서는 윈도우 제약 조건을 사용하기 때문에 질의 시퀀스 Q의 어느 한 요소는 데이타 시퀀스내의 정해진 영역내의 요소들하고만 대응되어질 수 있다. 이를 이용하여 데이타 시퀀스로부터 질의 시퀀스에 독립적인 특징을 추출할 수 있다.

먼저 데이타 시퀀스 S와 질의 시퀀스 Q를 정해진 개수 d개의 세그먼트로 나눈다. 이 때 데이타 시퀀스 S의 각 세그먼트 S_k는 전체 시퀀스의 $\frac{1}{d}$에 덧붙여 좌우로 w개의 요소를 더 포함하게 된다. 이는 윈도우의 크기 w를 고려한 것이다. 윈도우 크기가 w일 때 데이타 시퀀스 S와 질의 시퀀스 Q의 k번째 세그먼트가 포함하는 요소의 범위는 다음과 같다.

$$S_k=s_{\lfloor\frac{(k-1)|S|}{d}-w+1\rfloor}\ \cdots\ s_{\lfloor\frac{k|S|}{d}+w\rfloor}$$
$$Q_k=q_{\lfloor\frac{(k-1)|Q|}{d}+1\rfloor}\ \cdots\ q_{\lfloor\frac{k|Q|}{d}\rfloor}$$

예를 들어 w는 1이고 d가 4일 때 <그림 1>과 같이 데이타 시퀀스 S와 질의 시퀀스 Q는 각각 4개의 세그먼트로 분할된다.



<그림 1> 시퀀스를 세그먼트로 분할하는 예 (d=4).

시퀀스 Q의 각 요소는 윈도우 제약 조건에 의해서 대응될 수 있는 시퀀스 S의 요소의 범위가 정해진다. 시퀀스 Q의 세그먼트 Q_k의 각 요소들에 대하여 대응될 수 있는 S의 요소의 범위를 모두 찾아서 합집합을 구하면 시퀀스 S의 세그먼트 S_k와 일치하게 된다. 그러므로 Q의 세그먼트 Q_k에 속하는 모든 요소는 반드시 시퀀스 S의 대응되는 세그먼트 S_k에 속하는 요소와 대응되어야 한다. 이러한 사실을 바탕으로 시간 왜곡 변환 거리 D_{tw}의 하한 함수 D_{lb}를 정의할 수 있다.

정의 1: 두 시퀀스 S와 Q에 대하여, 시간 왜곡 변환 거리의 하한 함수는 다음과 같이 정의한다.

$$D_{lb}(S,Q)=L_\infty(\langle S_1,\cdots,S_k,\cdots,S_d\rangle,\langle Q_1,\cdots,Q_k,\cdots,Q_d\rangle)$$

두 시퀀스 S와 Q의 각 세그먼트들 간의 거리의 최대값을 하한 함수 D_{lb}로 정의한다. 이 때 데이타 시퀀스 S의 한 세그먼트 S_k와 질의 시퀀스의 세그먼트 Q_k사이의 거리는 다음과 같다.

정의 2: 세그먼트 S_k와 Q_k의 거리는 다음과 같이 정의한다.

$$|S_k-Q_k|=\max(D(S_k,\min(Q_k)),D(S_k,\max(Q_k)))$$

$$D(S_k,q_j)=\begin{cases}q_j-\max(S_k) & (q_j>\max(S_k))\\ 0 & (\max(S_k)\ge q_j\ge\min(S_k))\\ \min(S_k)-q_j & (\min(S_k)>q_j)\end{cases}$$

두 세그먼트 S_k와 Q_k사이의 거리는 Q_k의 최대값과 S_k간의 거리와 Q_k의 최소값과 S_k간의 거리 중 큰 값으로 정의한다. 이 때 세그먼트 S_k와 Q의 한 요소 q_j사이의 거리는 q_j가 S_k의 요소값들의 범위 안에 있을 때는 0이고 그렇지 않을 경우는 범위의 최대, 최소와의 차이 중에서 작은 값으로 정의한다.

하한 함수 D_{lb}로부터 다음의 정리를 이끌어 낼 수 있다.

정리 1

임의의 두 시퀀스 S, Q에 대해서 다음 식이 항상 성립한다.

$$D_{lb}(S, Q) \le D_{tw}(S, Q)$$

그러므로 임의의 허용치 ε에 대해 다음 식이 항상 성립한다.

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

정리 1에 의해서 유사 검색 질의를 처리할 때 D_{tw} 대신 D_{lb}를 사용하여도 착오 기각이 발생하지 않음을 알 수 있다. 그러므로 이를 이용하여 다차원 인덱스를 만들고 질의 시퀀스와 유사할 가능성이 큰 후보 시퀀스들을 골라낼 수 있다.

다차원 인덱스를 구성하는 방법은 다음과 같다. (1) 각각의 데이타 시퀀스를 d개의 세그먼트로 나눈다. (2) 각 세그먼트내의 요소들의 최대, 최소값을 구한다. (3) MBR의 각 차원의 범위를 (2)에서 구한 각 세그먼트의 요소값의 범위로 설정한다. 즉 d차원의 두 점 ($\min(S_1)$, …, $\min(S_k)$, …, $\min(S_d)$)와 ($\max(S_1)$, …, $\max(S_k)$, …, $\max(S_d)$)를 양 끝점으로 하는 MBR을 구성한다. (4) (3)에서 구성한 MBR을 다차원 인덱스에 삽입한다.

질의를 처리할 때는 질의 시퀀스 Q를 데이타 시퀀스를 처리할 때와 유사한 방법으로 d개의 세그먼트로 나누어 질의 MBR을 구성한 다음 다차원 인덱스를 검색하여 후보를 선택한다. 검색의 결과로 반환되는 후보 시퀀스들에 대해서 실제 거리 D_{tw}를 계산하여 최종 결과를 구한다.

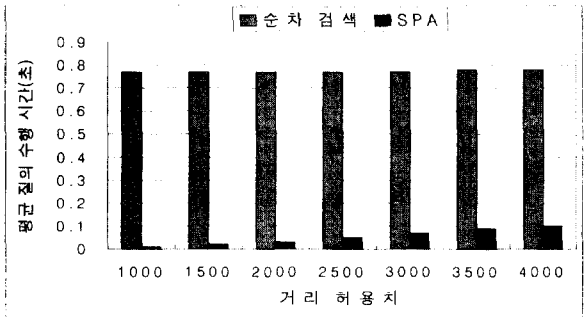
5. 성능 평가

이 절에서는 실제 주식 데이타를 이용한 실험을 통하여 SPA의 성능을 분석한다. 데이타 시퀀스의 평균 길이는 256이고 시퀀스의 개수는 1000개이다. 100번의 유사 시퀀스 검색을 수행하였을 때의 평균 질의 수행 시간과 후보 개수의 비율을 측정하였다.

다차원 인덱스로는 공간 객체(spatial object)를 저장할 수 있는 공간 색인 방법(spatial access methods)을 사용하는 기법은 모두 사용가능한데 이 실험에서는 R*-트리[7]를 이용하였다. MBR의 차원수는 8로 고정하였다. 즉, 데이타 시퀀스와 질의 시퀀스를 각각 8개의 세그먼트로 나누어서 특징을 추출하였다.

성능의 비교는 모든 데이타 시퀀스를 읽어 질의 시퀀스와 비교하는 순차검색과 이 논문에서 제안하는 세그먼트 분할 기법(SPA)을 비교하였다.

먼저, 허용치 ε을 변화시킬 때 실행시간을 측정해보았다. 윈도우 크기는 20으로 고정하였다. <그림 2>는 그 실험결과를 나타낸 것이다. 가로축은 허용치 ε을 나타내며, 세로축은 실행시간을 나타낸다.

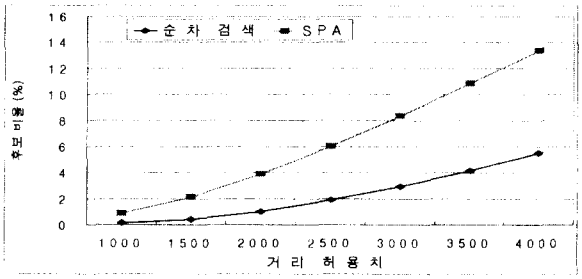


<그림 2> 평균 질의 수행 시간.

SPA가 순차 검색보다 약 8배에서 70배까지 더 나은 성능을 보였다. SPA가 전체 R*-tree의 5% 미만의 작은 영역만을 탐색하여 후보를 선택하기 때문에 인덱스를 이용한 필터링의 효과가 매우 좋다는 것을 알 수 있다.

<그림 3>은 SPA를 통하여 선택되는 후보의 수와 질의를 처리한 최종결과의 개수가 전체 데이타 시퀀스 개수와 비교하여 그 비율이 얼마인지를 나타낸다.

후보의 수는 최종결과의 약 2.5배에서 5배 정도로 나타났다. 허용치 ε이 작을 때, 즉 질의를 처리한 최종결과의 개수가 적을 때 SPA는 약 5% 미만의 후보를 선택하고 이들 후보에 대해서만 실제 거리 D_{tw}를 계산하게 된다. 그러므로 순차검색에 비해서 매우 좋은 성능을 나타낸다. 이는 실제 응용에서 요구하는 질의 결과의 수가 대부분의 경우 적다는 것을 고려할 때 매우 바람직한 성질이다.



<그림 3> 전체 시퀀스 중 후보의 비율.

6. 결론

이 논문에서는 윈도우 제약 조건을 가지는 시간 왜곡 변환 지원 유사 시퀀스 검색을 제안하였다. 세그먼트 분할 기법(SPA)은 데이타 시퀀스를 세그먼트로 분할하고 각 세그먼트의 범위를 이용하여 시간 왜곡 변환 거리의 하한 함수 D_{lb}를 정의하고 이를 기반으로 다차원 인덱스를 구성하여 질의 시퀀스와 유사한 후보를 빠르게 검색하고 마지막으로 해당 후보들에 대해서만 실제 D_{tw}를 계산하여 최종 결과를 구하기 때문에 전체 질의 처리 시간을 단축시킬 수 있다. 실험 결과에서 알 수 있듯이 SPA는 순차검색에 비하여 좋은 성능을 보인다. 특히 거리 허용치가 작을 때 더욱 우수한 성능을 보인다.

7. 참고 문헌

[1] R. Agrawal, C. Faloutsos, and A. Swami, *"Efficient similarity search in sequence databases,"* In Proc. of the FODO Conf., pp. 69-84, Oct. 1993.

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

[3] E. J. Keogh and M. J. Pazzani, *"Derivative Dynamic Time Warping,"* In First SIAM International Conference on Data Mining, 2001.

[4] B. K. Yi, H. V. Jagadish, and C. Faloutsos *"Efficient Retrieval of Similar Time Sequences Under Time Warping,"* In Proc. IEEE ICDE, pp. 201-208, 1998.

[5] 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 IEEE ICDE, pp. 607-614, 2001.

[6] S. H. Park, W. W. Chu, J. Yoon, C. Hsu, *"Efficient Searches for Similar Subsequences of Different Lengths in Sequence Databases,"* In Proc IEEE ICDE, pp. 23-32, 2000.

[7] N. Beckmann, H. Kriegel, R. Schneider, B. Seeger, *"The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles,"* In Proc. of the ACM SIGMOD Conf., pp. 322-331, 1990.