타임 워핑 하의 효율적인 시계열 서브시퀀스 매칭을 위한 접두어 질의 기법의 확장

장 병 철[†] · 김 상 욱[††] · 차 재 혁[†††]

요 약

본 논문에서는 타임 워핑 하의 시계열 서브시퀀스 매칭을 처리하는 방법에 대하여 논의한다. 타임 워핑은 시퀀스의 길이가 서로 다른 경우에도 유사한 패턴을 갖는 시퀀스들을 찾을 수 있도록 해 주는 변환이다. 접두어 질의 기법(prefix-querying method)는 착오 기각 없이 타임 워핑 하의 시계열 서브시퀀스 매칭을 처리하는 인덱스를 이용한 최초의 방식이다. 이 방법은 사용자가 질의를 편리하게 작성하도록 하기 위하여 기본 거리 함수로서 L_∞를 사용한다. 본 논문에서는 L_∞ 대신 타임 워핑 하의 시계열 서브시퀀스 매칭에서 기본 거리 함수로서 가장 널리 사용되는 L_1을 적용할 수 있도록 접두어 질의를 확장한다. 또한, 제안된 기법으로 타임 워핑 하의 시계열 서브시퀀스 매칭을 수행하는 경우 착오 기각(false dismissal)이 발생하지 않음을 이론적으로 증명한다. 다양한 실험을 통한 성능 평가를 통하여 본 연구에서 제시하는 기법의 우수성을 검증한다. 실험 결과에 의하면, 제안된 기법은 가장 좋은 성능을 보이는 기존의 기법과 비교하여 매우 뛰어난 성능 개선 효과를 보이는 것으로 나타났다.

키워드 : 유사검색, 서브시퀀스 매칭, 접두어 질의, 타임 워핑, 인덱싱 기법

On Extending the Prefix-Querying Method for Efficient Time-Series Subsequence Matching Under Time Warping

Byoungchol Chang[†] · Sang-Wook Kim[††] · Jaehyuk Cha[†††]

ABSTRACT

This paper discusses the way of processing time-series subsequence matching under time warping. Time warping enables finding sequences with similar patterns even when they are of different lengths. The *prefix-querying* method is the first index-based approach that performs time-series subsequence matching under time warping without false dismissals. This method employs the L_∞ as a base distance function for allowing users to issue queries conveniently. In this paper, we extend the prefix-querying method for absorbing L_1, which is the most-widely used as a base distance function in time-series subsequence matching under time warping, instead of L_∞. We also formally prove that the proposed method does not incur any false dismissals in the subsequence matching. To show the superiority of our method, we conduct performance evaluation via a variety of experiments. The results reveal that our method achieves significant performance improvement in orders of magnitude compared with previous methods.

Key Words : Similarity Search, Subsequence Matching, Prefix Querying, Time Warping, Indexing Technique

1. 서 론

시계열 데이터베이스(time-series database)란 객체의 변화되는 값들의 연속으로 구성된 데이터 시퀀스(data sequence)들의 집합이다[1]. 대표적인 예로는 주가 데이터, 환율 데이터, 기온 데이터, 제품 판매량 데이터, 기업 성장률 데이터 등이 있다[2-4]. 시퀀스 매칭(sequence matching)이란 주어진 질의 시퀀스(query sequence)와 변화의 패턴이 유사한 시퀀스들을 시계열 데이터베이스로부터 찾아내는 데이터 마이닝(data mining) 및 데이터 웨어하우징(data warehousing) 분야의 중요한 연산이다[1, 3-6].

시퀀스 매칭에 관한 대부분의 기존 연구에서는 길이 n의 시퀀스를 n 차원 공간상의 한 점으로 간주한다. 또한, 길이 n인 서로 다른 두 시퀀스 $X(=[x_1, x_2, ..., x_n])$와 $Y(=[y_1, y_2, ..., y_n])$ 간의 유사한 정도를 측정하는 척도로서 아래의 식과 같이 정의되는 거리 함수 $L_p(X,Y)$를 널리 사용한다. L_1은 맨하탄 거리(Manhattan distance), L_2는 유클리드 거리(Euclidean distance), L_∞은 대응되는 각 요소 값 쌍의 거리

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 지원산업의 연구결과로 수행되었음(IITA-2005-CIO90-0502-0009).
† 정 회 원 : 한양대학교 정보통신학과 박사과정
†† 종신회원 : 한양대학교 정보통신대학 정보통신학부 교수
††† 종신회원 : 한양대학교 정보통신대학 정보통신학부 부교수
논문접수 : 2005년 9월 8일, 심사완료 : 2006년 1월 6일

중 최대 거리를 의미한다[7]. 응용에서 주어진 허용치 ε보다 작거나 같은 $L_p(X,Y)$를 갖는 임의의 두 시퀀스 X, Y를 유사하다고 간주한다[1, 4, 6, 8-11].

$$L_p(X,Y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$$

L_p 거리 함수만을 이용한 시퀀스 매칭을 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. 초기의 연구인 참고 문헌[1, 4] 등에서는 변환을 지원하지 않았으나, 이후에는 스케일링(scaling)[3, 9], 시프팅(shifting)[3, 9], 정규화(normalization)[10, 12, 13], 이동 평균(moving average)[6, 14], 타임 워핑(time warping)[15-19] 등의 다양한 변환을 지원하는 방법들이 제안되었다.

이들 중 타임 워핑은 시퀀스내의 각 요소 값을 임의의 수만큼 반복시키는 것을 허용하는 변환이다[16]. 타임 워핑 후의 두 시퀀스들 간의 거리를 타임 워핑 거리(time warping distance)라 한다. 두 시퀀스 S와 Q간의 타임 워핑 거리(time warping distance) D_{tw}는 다음과 같이 재귀적으로 정의된다[16-18]:

[정의 1]

(1) $D_{tw}(<>, <>) = 0$,

(2) $D_{tw}(S, <>) = D_{tw}(<>, Q) = \infty$,

(3) $D_{tw}(S, Q) = (|L_p(First(S), First(Q))|^p + |\min(D_{tw}(S, Rest(Q)), D_{tw}(Rest(S), Q), D_{tw}(Rest(S), Rest(Q)))|^p)^{1/p}$

여기서, First(S)는 S의 첫 번째 요소 s_1을 의미하며, Rest(S)는 s_1을 제외한 S의 나머지 요소들로 구성되는 시퀀스를 의미한다. <>은 요소가 존재하지 않는 널 시퀀스(null sequence)를 의미한다. min은 세 개의 인자들 중 가장 작은 값을 가지는 것을 취하는 함수이다. L_p는 응용에서 적합한 것을 선택하여 사용할 수 있으나, 현재 맨해튼 거리(Manhattan distance) L_1을 기반으로 하는 타임 워핑 거리가 가장 널리 사용되고 있다.

두 시퀀스들을 대상으로 하는 타임 워핑은 변환 후의 두 시퀀스들 간의 타임 워핑 거리를 최소화하는 방향으로 진행된다. 예를 들어, 두 시퀀스 S = <20, 21, 21, 20, 20, 23, 23, 23>와 Q = <20, 20, 21, 20, 23>은 타임 워핑에 의하여 동일한 시퀀스 <20, 20, 21, 21, 20, 20, 23, 23, 23>으로 변환될 수 있으며, 이 결과 $D_{tw}(S, Q)$는 0이 된다.

전술한 바와 같이, L_p 거리 함수만을 이용한 시퀀스 매칭을 통해서는 두 시퀀스의 길이가 동일한 경우에만 적용할 수 있다. 반면, 타임 워핑 거리는 데이터베이스내의 시퀀스들의 길이가 서로 달라서 L_p 거리 함수를 이용하여 유사 정도를 직접 측정할 수 없는 경우에 매우 유용하다[18][1]. 현재, 타임 워핑은 음성 인식 분야에서 널리 사용되고 있으며 [20], 심전도 데이터, 주가 데이터, 기온 데이터, 기업 성장률 데이터 등에도 유사한 방식으로 적용할 수 있다.

저자가 참여한 참고 문헌 [18, 21]에서는 인덱스 기반의 타임 워핑하의 시퀀스 매칭 기법을 제안한 바 있고, 참고 문헌 [19]에서는 이를 확장한 서브시퀀스 매칭 기법인 접두어 질의 기법(prefix-querying method)를 제안한 바 있다. 이 두 기법들은 사용자의 질의 편리성을 위하여 L_∞을 기본 거리 함수로 사용한다. 그러나 국내외 학술회의 또는 학술지를 통한 논의 및 심사과정에서 본 저자들은 L_1을 위한 인덱스 기반 타임 워핑하의 시퀀스 매칭 기법의 확장에 대한 향후 연구에 대하여 요청을 받은 바 있다[22]. 본 논문은 이러한 요청에 대한 답으로서 작성되었다.

본 논문에서는 타임 워핑을 지원하는 인덱스 기반 서브시퀀스 매칭 기법에 관하여 논의하고자 한다. 기존 접두어 질의 기법[19]이 거리 함수 L_1을 적용하는 경우에도 올바르게 동작 하는가의 여부는 현재까지 검증된 바 없다. 본 논문에서는 L_1을 적용할 수 있도록 접두어 질의 기법을 확장하는 방안을 제시한다. 또한 확장된 기법이 착오 기각(false dismissal)없이 검색 대상이 되는 모든 서브시퀀스들을 올바르게 검색한다는 것을 이론적으로 증명한다. 제안된 기법의 우수성을 규명하기 위하여 기존의 기법들과 실험을 통한 성능 평가를 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 타임 워핑 지원 유사 검색에 관한 기존의 연구에 관하여 소개하고, 3장에서는 접두어 질의 기법의 확장 방안에 대하여 논의한다. 4장에서는 타임 워핑 하의 서브시퀀스 매칭을 위한 기존 기법들과 접두어 질의 기법에 대한 성능 평가 결과를 제시한다. 끝으로, 5장에서는 본 논문을 요약하고 결론을 내린다.

2. 관련 연구

본 장에서는 관련 연구로서 타임 워핑 하의 시퀀스 매칭을 착오 기각 없이 수행하는 기존의 연구들인 Naïve-Scan, LB-Scan, ST-Filter, LB-Filter와 접두어 질의 기법을 소개한다. 각 기법에 관하여 (1) 전체 매칭 방안, (2) 서브시퀀스 매칭 방안[2], (3) 특징에 관하여 논의한다.

2.1 Naïve-Scan[15]

전체 매칭 방안: 디스크로부터 각 데이터 시퀀스를 액세스한 후, 이 데이터 시퀀스 S와 질의 시퀀스 Q의 타임 워핑

1) 다음과 같은 경우, 서로 다른 길이를 가지는 시퀀스들 간의 유사 정도의 측정이 요구된다[18]. 첫째, 두 시퀀스들을 위한 요소 값의 측정 주기가 다른 경우이다. 예를 들어, 한 시퀀스는 매 분마다 요소 값을 측정하고, 다른 시퀀스는 매 시간마다 요소 값을 측정할 수 있다. 둘째, 측정 주기는 같지만, 측정 시작 시점이 다른 경우이다. 예를 들어, 한 시퀀스는 측정이 1년 전부터 시작되었지만, 다른 시퀀스는 새로 데이터베이스에 추가되어 오늘부터 측정이 시작될 수 있다. 이와 같이 비교하고자 하는 시퀀스들의 길이가 서로 다른 경우, 타임 워핑은 시퀀스들의 개별적인 요소 값의 차이보다는 시간의 변화에 따르는 시퀀스들의 전체적인 경향이 얼마나 유사한가를 파악하는데 유용하게 사용되는 변환이다.

2) ST-Filter는 원래 서브시퀀스 매칭을 대상으로 제안된 기법인 반면, Naïve-Scan과 LB-Scan은 전체 매칭을 대상으로 제안된 기법이다. 본 장에서는 Naïve-Scan과 LB-Scan을 기본 아이디어를 이용하여 서브시퀀스 매칭을 수행하는 일반적인 방안을 소개한다.

거리 $D_{tw}(S, Q)$를 계산함으로써 전체 매칭을 수행한다. $D_{tw}(S, Q)$를 효과적으로 계산하기 위한 방법으로서 동적 프로그래밍(dynamic programming)을 사용한다[15]. 계산된 $D_{tw}(S, Q)$ 값이 허용치 ε이하인 경우, 해당 시퀀스 S가 질의 시퀀스 Q와 유사하다고 간주한다.

동적 프로그래밍을 사용하여 S와 Q간의 타임 워핑 거리를 계산할 때, 거리 축적 테이블(cumulative distance table) T의 각 요소 $T(i,j)$[3]는 다음과 같은 재귀 관계(recurrence relation)에 의하여 구성된다[15]. 동적 프로그래밍 알고리즘은 아래의 재귀 관계를 이용하여 거리 축적 테이블 T를 아래에서 위로 채워나간다.

$$T(0, 0) = 0$$
$$T(0, j) = T(i, 0) = \infty$$
$$T(i, j) = |Q[i] - S[j]| + min(T(i-1, j), T(i, j-1), T(i-1, j-1))$$

다음의 (그림 1)은 기본 거리 함수로서 L_1이 사용되는 경우, 거리 축적 테이블을 이용한 두 시퀀스 S와 Q의 타임 워핑 거리 계산 예를 보인다. 계산 결과, $D_{tw}(S, Q)$는 12가 된다.

6	16	11	**12**
6	13	**9**	10
7	10	**7**	8
6	6	**4**	5
5	3	**2**	3
4	1	1	2
S \ Q	3	4	3

(그림 1) L_1을 이용한 S = ⟨4, 5, 6, 7, 6, 6⟩과 Q = ⟨3, 4, 3⟩의 타임 워핑 거리 계산의 예

서브시퀀스 매칭 방안: 각 데이터 시퀀스 S를 디스크로부터 액세스한 후, S에 속하는 각 서브시퀀스 S[i : j]에 대하여 질의 시퀀스 Q와의 타임 워핑 거리 $D_{tw}(S[i : j], Q)$를 동적 프로그래밍을 이용하여 계산함으로써 서브시퀀스 매칭을 수행한다. 참고 문헌 [23]에서는 모든 서브시퀀스가 아닌 모든 접미어 S[i : |S|]에 대해서만 거리 축적 테이블을 구성하는 방식을 사용하여 계산 과정의 중복을 제거하는 기법을 제안하였다.

특징: 여과 단계를 거치지 않으므로 후처리 단계의 수행 시간이 지나치게 크다[18]. 즉, 모든 데이터 시퀀스들을 디스크로부터 액세스해야 한다는 부담이 있다. 또한, 기본적인 Naïve-Scan 방식을 사용할 경우 (서브)시퀀스 S와 Q의 D_{tw}를 계산할 때의 CPU 수행 시간은 O(|S|*|Q|)이므로 매우 크다. 여기서, |S|와 |Q|는 각각 시퀀스 S와 Q의 크기를 의미한다. 이 결과, 많은 시퀀스들로 구성되는 대형 데이터베이스 환경에서는 검색 성능이 떨어진다.

3) 본 연구에서는 거리함수 L_1에 대해서만 논하므로 축적 테이블을 구성하는 공식 T도 L_1만 고려한다.

2.2 LB-Scan[16]

전체 매칭 방안: 타임 워핑 거리 D_{tw}의 반환 값보다 항상 작은 값을 반환하는 하한 함수(lower-bound function) D_{lb}를 이용하여 여과 단계를 수행한다. 즉, 여과 단계에서는 디스크로부터 각 데이터 시퀀스를 액세스한 후, 이 데이터 시퀀스 S와 질의 시퀀스 Q에 대하여 $D_{lb}(S, Q)$를 적용한다. 여과 단계에서 D_{lb}의 반환 값이 허용치 ε이하인 데이터 시퀀스 S'에 대해서는 질의 시퀀스 Q와의 타임 워핑 거리 $D_{tw}(S', Q)$를 계산하는 후처리 단계를 수행한다. $D_{tw}(S', Q)$의 계산을 위하여 Naïve-Scan과 동일한 방식으로 동적 프로그래밍을 사용한다.

서브시퀀스 매칭 방안: 여과 단계에서 디스크로부터 각 데이터 시퀀스 S를 액세스한 후, S에 속하는 각 서브시퀀스 S[i : j]와 질의 시퀀스 Q에 대하여 $D_{lb}(S[i : j], Q)$를 적용한다. 여기서에서 D_{lb}의 반환 값이 허용치 ε이하인 서브시퀀스 S[i : j]'에 대해서는 동적 프로그래밍을 이용하여 질의 시퀀스 Q와의 타임 워핑 거리 $D_{tw}(S[i : j]', Q)$를 계산하는 후처리 단계를 수행한다.

특징: 별도의 자료 구조를 채택하지 않으므로 여과 단계에서 모든 데이터 시퀀스들이 디스크로부터 액세스된다. 따라서 디스크 액세스 시간은 Naïve-Scan과 동일하다. 여과 단계에서는 모든 서브시퀀스 S와 질의 시퀀스 Q 간의 D_{lb}가 계산된다. 각 $D_{lb}(S, Q)$를 계산할 때의 CPU 수행 시간은 O(|S|+|Q|)로서 O(|S|*|Q|)인 $D_{tw}(S, Q)$와 비교하여 CPU 수행 시간이 매우 작다[7]. 여과 단계를 통하여 최종 결과에 포함될 가능성이 없는 (서브)시퀀스들을 사전에 제외시킬 수 있으므로 후처리 단계의 수행 시간을 크게 줄일 수 있다[18]. 따라서 여과 단계에서 제외되는 (서브)시퀀스들이 많은 경우, 성능 개선 효과는 매우 크다.

2.3 ST-Filter[17]

전체 매칭 방안: 여과 단계를 위하여 데이터베이스 내의 각 시퀀스의 요소 값들을 심벌로 변환시키고, 이들을 접미어 트리(suffix tree)[24] 내에 저장시킨다. 여과 단계에서는 접미어 트리 검색을 이용하여 질의 시퀀스 Q와의 타임 워핑 거리 D_{tw}가 허용치 ε이하일 가능성이 있는 후보 시퀀스 S'들을 걸러낸다. 후처리 단계에서는 이러한 각 S'을 대상으로 동적 프로그래밍을 사용하여 $D_{tw}(S', Q)$를 계산한다.

서브시퀀스 매칭 방안: 여과 단계를 위하여 각 데이터 시퀀스 내 각 접미어의 요소 값들을 심벌로 변환시키고, 이들을 접미어 트리 내에 저장시킨다. 여과 단계에서는 접미어 트리 검색을 통하여 질의 시퀀스 Q와의 타임 워핑 거리 D_{tw}가 허용치 ε이하일 가능성이 있는 후보 서브시퀀스 S[i : j]'들을 걸러낸다. 후처리 단계에서는 이러한 각 S[i : j]'을 대상으로 $D_{tw}(S[i : j]', Q)$를 계산한다.

특징: 접미어 트리 검색을 사용하므로 LB-Scan과는 달리 전체 데이터 시퀀스들이 아닌 접미어 트리의 일부만을 디스크로부터 액세스함으로써 여과 단계를 수행할 수 있다. 그러나 이 접미어 트리의 크기는 데이터 시퀀스들이 저장된 파일보다 큰 것이 일반적이다. 또한, 좋은

시퀀스 매칭 성능을 제공하기 위한 최적의 도메인 분류(categorization)[17]가 쉽지 않으며[18], 동일한 데이터베이스의 경우에도 질의 시퀀스마다 서브시퀀스 처리 성능에 큰 차이가 있다.

2.4 LB-Filter[18]

전체 매칭 방안: 질의 처리 성능 개선을 위하여 다차원 인덱스를 이용한다. 질의 시퀀스, 데이터 시퀀스 각각에서 4개의 요소, First, Last, Greatest, Smallest 추출하고 이를 특성 벡터로 정의한다. 데이터 시퀀스의 특성 벡터를 사차원 트리로 구성하고, 질의 시퀀스의 특성 벡터를 이용한 트리 탐색을 통하여 후보 시퀀스 S′들을 걸러낸다. 후처리 단계에서는 이러한 각 S′를 대상으로 동적 프로그래밍을 사용하여 $D_{tw}(S', Q)$를 계산한다.

서브시퀀스 매칭 방안: LB-Filter를 이용한 서브시퀀스 매칭을 접두어 질의(prefix-querying)라고 한다[19]. 접두어 질의는 데이터 시퀀스로부터 미리 정해진 크기의 윈도우를 추출하고, 각 윈도우로부터 전체 매칭에서와 같이 특성 벡터를 추출한 후, 이들을 대상으로 사차원 트리를 구성한다. 질의 처리 시에는 질의 시퀀스로부터 데이터 시퀀스와 매치될 수 있는 접두어들을 추출한 후, 각 접두어로부터 특성 벡터를 추출한다. 각 질의 특성 벡터를 이용한 트리 탐색을 통하여 후보 시퀀스 S′들을 걸러낸다. 후처리 단계에서는 이러한 각 S′를 대상으로 동적 프로그래밍을 사용하여 $D_{tw}(S', Q)$를 계산한다.

특징: 다차원 인덱스를 사용하므로 LB-Scan과는 달리 여과 단계를 빠르게 수행할 수 있다. 그리고 ST-Filter에서 사용하는 트리는 그 크기가 데이터 시퀀스의 크기보다 커지는 반면, 접두어 질의는 네 개의 특성 벡터를 이용하므로 트리 용량도 크지 않다. 또한 착오 기각 없이 허용치 내의 모든 시퀀스를 찾을 수 있으므로 후처리 시간을 줄이는 동시에 정확도도 높일 수 있다.

3. 접두어 질의의 확장

본 장에서는 기본 거리로서 L_1을 사용할 수 있도록 참고문헌 [19]의 접두어 질의 기법을 확장하고자 한다. 참고 문헌 [18]에서 LB-Filter 기법의 D_{tw_lb}가 D_{tw}의 하한 함수인 동시에 유사 검색에서 사용하는 거리 함수 L_∞에 대해 삼각 부등식을 만족함을 보임으로써 착오 기각이 발생하지 않음을 증명하였다. 본 연구에서는 이를 확장하여 거리 함수 L_1에 대해서도 접두어 질의 기법을 적용할 수 있음을 증명하고자 한다.

[정의 2]

$$D_{tw\_lb}(S, Q) = L_1(\text{Feature}(S), \text{Feature}(Q))$$

여기서 Feature(S) = <First(S), Last(S), Greatest(S), Smallest (S)>, Feature(Q) = <First(Q), Last(Q), Greatest(Q), Smallest (Q)> 이다.

다음에는 [정리 1]과 [정리 2]를 이용하여 함수 D_{tw_lb}가 타임 워핑 거리 D_{tw}의 하한 함수인 동시에 삼각 부등식을 만족함을 보이고자 한다. [정리 1]의 증명을 위하여 다음의 [보조 정리 1]과 [보조 정리 2] 및 [가정 1]을 이용한다. [가정 1]은 본 논문의 [정리 3]과 그것의 [따름 정리 2]까지 적용된다. [정리 5]에서는 [가정 1]이 성립하지 않는 경우에 대한 해결 방안을 제시 한다.

[가정 1] 데이터 시퀀스 S와 질의 시퀀스 Q의 워핑된 시퀀스를 S′, Q′라 하면, $S' = \langle s_1', s_2', \ldots, s_k' \rangle$, $Q' = \langle q_1', q_2', \ldots, q_k' \rangle$에서 s_1', s_k', q_1', q_k'은 각 시퀀스 내에서 최대값 또는 최소값이 아니다.

[보조 정리 1] 임의의 두 시퀀스 $S = \langle s_1, s_2, \ldots, s_n \rangle$, $Q = \langle q_1, q_2, \ldots, q_m \rangle$에 대하여 다음이 항상 성립한다.

$$D_{tw}(S, Q) \geq L_1(\langle \text{First}(S), \text{Last}(S) \rangle, \langle \text{First}(Q), \text{Last}(Q) \rangle)$$

[증명] 시퀀스 S와 Q의 타임 워핑이란 S와 Q를 요소 반복을 통하여 최소의 L_p를 갖는 시퀀스들로 상호 변환하는 것이다. 이 변환된 시퀀스를 S′과 Q′이라 하고, $|S'|=|Q'|=k$ ($|S| \leq k$, $|Q| \leq k$)이라 하자.

$$\begin{aligned} D_{tw}(S, Q) &= L_1(S', Q') \\ &= L_1(\langle s_1', s_2', \ldots, s_k' \rangle, \langle q_1', q_2', \ldots, q_k' \rangle) \\ &= L_1(\langle s_2', s_3', \ldots, s_{k-1}' \rangle, \langle q_2', q_3', \ldots, q_{k-1}' \rangle) + \\ &\quad L_1(\langle s_1', s_k' \rangle, \langle q_1', q_k' \rangle) \\ &= L_1(\langle s_2', s_3', \ldots, s_{k-1}' \rangle, \langle q_2', q_3', \ldots, q_{k-1}' \rangle) \\ &\quad + L_1(\langle \text{First}(S), \text{Last}(S) \rangle, \langle \text{First}(Q), \text{Last}(Q) \rangle) \\ &\geq L_1(\langle \text{First}(S), \text{Last}(S) \rangle, \langle \text{First}(Q), \text{Last}(Q) \rangle) \end{aligned}$$

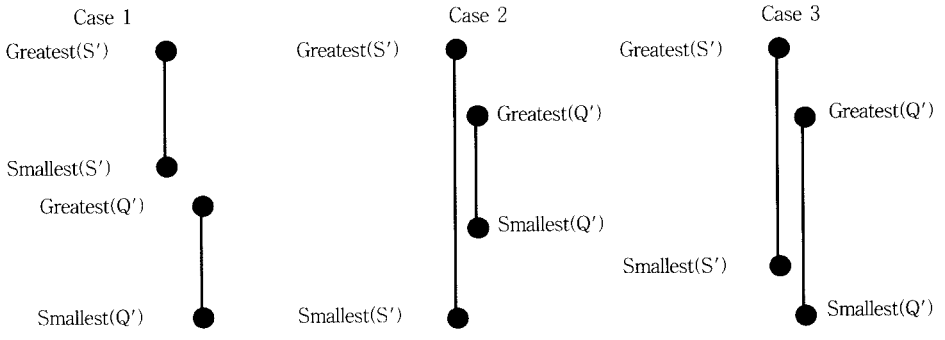
따라서 [보조 정리 1]은 항상 성립한다.

[보조 정리 2] 임의의 두 시퀀스 $S = \langle s_1, s_2, \ldots, s_n \rangle$, $Q = \langle q_1, q_2, \ldots, q_m \rangle$에 대하여 다음이 항상 성립한다.

$$D_{tw}(S, Q) \geq L_1(\langle \text{Greatest}(S), \text{Smallest}(S) \rangle, \langle \text{Greatest}(Q), \text{Smallest}(Q) \rangle)$$

[증명] [보조 정리 1]에서와 같이 시퀀스 S와 Q의 타임 워핑 변환된 시퀀스를 S′과 Q′라 하자. 또한, 변환 후 Greatest (S′), Smallest(S′)과 매치되는 Q′의 요소를 각각 Greatest_ Match (Q′), Smallest_Match(Q′)이라 정의하자. 같은 방식으로 변환 후 Greatest(Q′), Smallest(Q′)과 매치되는 S′의 요소를 각각 Greatest_Match(S′), Smallest_Match(S′)이라 정의하자.

본 증명에서는 (그림 2)에서 나타난 바와 같이 두 시퀀스의 요소 값의 분포 범위의 관계에 따라 발생 가능한 세 가지 경우에 대하여 본 보조 정리가 성립함을 보이고자 한다. 논의 전개의 편의상 Greatest(S′) ≥ Greatest(Q′)라 가정한다. 반대의 경우에는 본 증명에서 사용된 S와 Q의 역할을 바꾸기만 하면 된다.



(그림 2) 시퀀스 S'와 Q'의 요소 값 분포 범위의 관계

Case 1:S'과 Q'의 범위가 전혀 겹치지 않는 경우

$D_{tw}(S, Q) = L_1(S', Q')$
$\geq |Greatest(S') - Greatest\_Match(Q')| + |Smallest(Q') - Smallest\_Match(S')|$
$\geq |Greatest(S') - Greatest(Q')| + |Smallest(Q') - Smallest(S')|$
$= L_1(<Greatest(S'), Smallest(S')>, <Greatest(Q'), Smallest(Q')>)$
$= L_1(<Greatest(S), Smallest(S)>, <Greatest(Q), Smallest(Q)>)$

Case 2:S' 범위가 Q'의 범위를 포함하는 경우

$D_{tw}(S, Q) = L_1(S', Q')$
$\geq |Greatest(S') - Greatest\_Match(Q')| + |Smallest(S') - Smallest\_Match(Q')|$
$\geq |Greatest(S') - Greatest(Q')| + |Smallest(S') - Smallest(Q')|$
$= L_1(<Greatest(S'), Smallest(S')>, <Greatest(Q'), Smallest(Q')>)$
$= L_1(<Greatest(S), Smallest(S)>, <Greatest(Q), Smallest(Q)>)$

Case 3:S'과 Q'의 범위가 일부 겹치는 경우

$D_{tw}(S, Q) = L_1(S', Q')$
$\geq |Greatest(S') - Greatest\_Match(Q')| + |Smallest(Q') - Smallest\_Match(S')|$
$\geq |Greatest(S') - Greatest(Q')| + |Smallest(Q') - Smallest(S')|$
$= L_1(<Greatest(S'), Smallest(S')>, <Greatest(Q'), Smallest(Q')>)$
$= L_1(<Greatest(S), Smallest(S)>, <Greatest(Q), Smallest(Q)>)$

위와 같이 가능한 모든 경우에 대하여 성립하므로 보조 [정리 2]는 항상 성립한다.

[정리 1] 임의의 두 시퀀스 $S = <s_1, s_2, ..., s_n>$, $Q = <q_1, q_2, ..., q_m>$에 대하여 다음이 항상 성립한다.

$$D_{tw}(S, Q) \geq D_{tw\_lb}(S, Q)$$

[증명] 시퀀스 S와 Q의 타임 워핑 변환된 시퀀스를 S'과 Q'라 하자.

$D_{tw\_lb}(S, Q) = L_1(Feature(S), Feature(Q))$
$= L_1(<First(S), Last(S), Greatest(S), Smallest(S)>, <First(Q), Last(Q), Greatest(Q), Smallest(Q)>)$
$= L_1(<First(S), Last(S)>, <First<Q, Last(Q)>) + L_1(<Greatest(S), Smallest(S)>, <Greatest(Q), Smallest(Q)>)$

$D_{tw}(S, Q) = L_1(S', Q')$
$= L_1(<s_1', s_2', ..., s_k'>, <q_1', q_2', ..., q_k'>)$
$= L_1(<First(S), Last(S)>, <First(Q), Last(Q)>) + L_1(<s_2', s_3', ..., s_{k-1}'>, <q_2', q_3', ..., q_{k-1}'>)$

가정 1에 의하여 s_1', s_k', q_1', q_k'이 최대값 또는 최소값이 아니므로 [보조 정리 2]에 의해

$$L_1(<s_2', s_3', ..., s_{k-1}'>, <q_2', q_3', ..., q_{k-1}'>) \geq L_1(<Greatest(S), Smallest(S)>, <Greatest(Q), Smallest(Q)>)$$
$$\therefore D_{tw}(S, Q) \geq D_{tw\_lb}(S, Q)$$

[정리 1]을 이용하여 다음의 따름 [정리 1]을 쉽게 유도해 낼 수 있다.

[따름 정리 1] 임의의 두 시퀀스 $S = <s_1, s_2, ..., s_n>$, $Q = <q_1, q_2, ..., q_m>$과 임의의 값 ε에 대하여 다음이 항상 성립한다.

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

[정리 2] 임의의 세 시퀀스 X, Y, Z에 대하여 다음이 항상 성립한다.

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

[증명] $D_{tw\_lb}(S, Q) = L_p(Feature(S), Feature(Q))$ 이며, 거리 함수 L_p는 항상 삼각 부등식 성질을 만족하므로 [정리 2]는 항상 성립한다.

다음은 접두어 질의 기법을 위한 하한 함수 D_{tw_lb}에 기본 거리 함수 L_1을 적용하는 경우에도 착오 기각이 발생하지

않음에 대해 증명한다.

[정리 3] 임의의 두 시퀀스 s, q, 그리고 임의의 양수 w($1 \le w \le |s|$)에 대하여, 만일 s와 q의 타임 워핑 거리가 ε이내이면, s의 접두어 s[1 : w]와의 타임 워핑 거리가 ε이내인 q의 접두어가 반드시 존재한다. 즉, 아래의 공식이 성립한다.

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

[증명] $p=<p[1], p[2], ..., p[|p|]>$를 두 시퀀스 s와 q의 D_{tw}를 최소화하는 워핑 경로라 하자. 여기서 $|s| \le |p|$이고, $|q| \le |p|$이다. 또한, 워핑 경로의 각 요소를 $p[h]=(s[i_h], q[j_h])$라 하자. 또한 워핑 경로 내에 존재하는 s와 q의 타임 워핑된 시퀀스를 각각 $s'(=<s[i_h]>)$, $q'(=q[j_h])$라 하자. 여기서, $1 \le h \le |p|$, $1 \le ih \le |s|$, $1 \le jh \le |q|$이다. 이때, 두 시퀀스 s와 q의 D_{tw}는 다음과 같이 계산된다.

$$D_{tw}(s, q) = L_1(s', q')$$

여기서, 워핑 경로의 단조(Monotonicity) 및 연속(Continuity) 성질[15]에 따라 다음의 식이 만족된다.

$(\exists x)\ (p[x] = (s[w], q[j_x]))$, 여기서 $1 \le w \le |s|$, $1 \le x \le |p|$, $1 \le j_x \le |q|$

L_∞를 사용하는 D_{tw}는 워핑 경로 내에 존재하는 두 요소 값의 차의 최대 값을 취하며, L_1을 사용하는 D_{tw}는 두 요소 값의 차들의 합을 취하기 때문에 p[1]부터 p[x]까지의 서브 워핑 경로는 워핑 경로 p보다 항상 작거나 같은 타임 워핑 거리를 반환한다. 따라서 워핑 경로 p가 ε이하의 거리를 반환하는 경우, p[1]부터 p[x]까지의 서브 워핑 경로도 역시 ε이하의 거리를 반환한다. p[1]부터 p[x]까지의 서브 워핑 경로의 거리가 $D_{tw}(s[1:w], q[1:j_x])$를 의미하므로 위의 정리는 성립한다.

LB-Filter에서 사용하였던 거리 함수 D_{tw_lb}는 타임 워핑 거리 D_{tw}의 하한 함수이므로, [정리 3]으로부터 다음과 같은 따름 [정리 2]을 쉽게 유도할 수 있다.

[따름 정리 2] 임의의 두 시퀀스 s, q, 그리고 임의의 양수 w($1 \le w \le |s|$)에 대하여, 다음의 식이 항상 성립한다.

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

특성 벡터 내 다중 역할 요소 문제

[정리 1]의 증명에서 워핑된 시퀀스 $S' = <s_1', s_2', ..., s_k'>$, $Q' = <q_1', q_2', ..., q_k'>$에서 s_1', s_k', q_1', q_k'이 최대값 또는 최소값이 아니라고 전제하였다. 그러나 흔하지 않은 경우이지만 실제 응용에서는 s_1', s_k', q_1', q_k'이 최대값 또는 최소값이 되는 경우가 발생될 수 있다. 여기서는 이 문제에 대하여 고찰하고, 해결 방안을 제시한다.

워핑된 시퀀스 $S' = <s_1', s_2', ..., s_k'>$에서 s_1', s_k'이 최대값 또는 최소값이 된다는 것은 Feature(S) = <First(S), Last(S), Greatest(S), Smallest(S)>에서 Fisrt(S) = Greatest(S) or Smallest(S) 이거나 Last(S) = Greatest(S) or Smallest(S)가 된다는 것을 의미한다. 이는 질의 시퀀스에도 마찬가지로 적용된다. 이런 경우는 다음과 같이 2가지로 구분하여 생각할 수 있다.

Case 1: s_1', s_k'와 동일한 값이 s_2', s_3', ..., s_{k-1}'에 존재하는 경우

이 경우는 시퀀스의 첫 값(s_1')이나 끝 값(s_k')이 최대값 혹은 최소값 이기는 하나 동일한 값을 가지는 요소가 s_2', s_3', ..., s_{k-1}'에 별도로 존재하는 경우이다. 이 경우는 비록 s_1', s_k'이 최대값 또는 최소값이 되어도 최대값, 최소값에 해당되는 요소가 나머지 시퀀스에 존재하므로 특성벡터의 4가지 요소로서 시퀀스 내의 서로 다른 독립적인 요소를 지정한 셈이 된다. 따라서 [정리 1]의 증명에서 사용한 아래 식을 만족하여 문제가 발생하지 않는다.

$L_1(<s_2', s_3', ..., s_{k-1}'>, <q_2', q_3', ..., q_{k-1}'>)$
$\ge L_1(<Greatest(S), Smallest(S)>, <Greatest(Q), Smallest(Q)>)$

Case 2: s_1', s_k'와 동일한 값이 s_2', s_3', ..., s_{k-1}'에 존재하지 않는 경우

이 경우는 시퀀스의 첫 값이나 마지막 값이 최대값이거나 최소값이면서 워핑된 시퀀스에서 첫값과 마지막값을 제외한 나머지 시퀀스에서는 나타나지 않는 경우이다. 이런 경우 시퀀스에서 특성 벡터를 구성하면 둘 이상의 특성으로 표현되는 요소가 포함된 특성 벡터가 만들어지며 이 요소를 다중 역할 요소(multi-role element)라 정의 한다. 특성 벡터 내에 다중 역할 요소가 있다는 것은 특성 벡터의 두 특성이 워핑된 시퀀스내의 같은 한 요소로부터 추출 되었다는 것을 의미한다. 따라서 Greatest(S) 혹은 Smallest(S)가 $<s_2', s_3', ..., s_{k-1}'>$에 존재하지 않으므로 [정리 1]의 증명 식을 보장할 수 없다.

[예제 1] 특성 벡터 내에 다중 역할 요소가 존재함으로써 $D_{tw} < D_{tw\_lb}$ 현상이 발생하는 경우

S = <100, 20, 15, 5, 30>, Q = <15, 20, 15, 5, 30>
Feature(S) = <100, 30, 100, 5>, Feature(Q) = <15, 30, 30, 5>
S' = <100, 20, 15, 5, 30>, Q' = <15, 20, 15, 5, 30>
D_{tw} = 85, D_{tw_lb} = 155 $\quad\therefore D_{tw} < D_{tw\_lb}$

[정리 4] 접두어 질의가 L_1에 대하여 삼각 부등식을 만족하여 착오 기각이 발생하지 않으려면 임의의 시퀀스 S의 특성 벡터 Feature(S) = <First(S), Last(S), Greatest(S), Smallest(S)>를 구성하는 네 요소들은 워핑된 시퀀스 $S' = <s_1', s_2', s_3', ..., s_{k-1}', s_k'>$의 서로 다른 요소들로 구성되어야 한다.

[증명] 독립적으로 추출할 수 있다는 것은 $First(S)=s_1'$, $Last(S)=s_k'$라고 했을 때, $s_2', s_3', ..., s_{k-1}'$에서 Greatest(S), Smallest(S)를 추출할 수 있다는 것을 의미한다. 이 조건이

질의에 대해서도 만족한다면 이 조건에 의해 추출된 특성 벡터는 [정리 1]의 증명을 만족하므로 [정리 4]는 증명된다.

본 연구에서는 접두어 질의 기법에 기본 거리 함수로서 L_1을 적용하기 위하여 [정리 4]를 만족하지 못하는 [예제 1]과 같은 경우를 해결하는 방안을 제시한다. 이 방법은 질의 시 주어지는 ε에 일정 값을 곱하여 넓혀진 범위를 대상으로 인덱스 검색을 수행함으로써 착오 기각을 제거하는 방법이다.

[정리 5] 데이터 시퀀스 S 또는 질의 시퀀스 Q의 특성 벡터에 다중 역할 요소가 포함되어 있을 때, 2배의 ε값을 기준으로 질의를 수행하면 착오 기각이 발생하지 않는다.

[증명] [보조 정리 1]과 2에 의하면, 임의의 두 시퀀스 $S = <s_1, s_2, ..., s_n>$, $Q = <q_1, q_2, ..., q_m>$에 대하여 다음 두 식이 항상 성립한다.

$$D_{tw}(S, Q) \geq L_1(<First(S), Last(S)>, <First(Q), Last(Q)>)$$
$$D_{tw}(S, Q) \geq L_1(<Greatest(S), Smallest(S)>, <Greatest(Q), Smallest(Q)>)$$

$D_{tw_lb}(S, Q)$는 [정리 1]에서 보여준 것처럼 다음과 같이 정의할 수 있다.

$$\begin{aligned} D_{tw\_lb}(S, Q) &= L_1(Feature(S), Feature(Q)) \\ &= L_1(<First(S), Last(S)>, <First(Q), Last(Q)>) + \\ &\quad L_1(<Greatest(S), Smallest(S)>, <Greatest(Q), Smallest(Q)>) \end{aligned}$$

[보조 정리 1], [보조 정리 2], [정리 1]에서 사용된 증명식을 종합하면 특성 벡터 내에 다중 역할 요소의 존재 유무와 관계없이 다음 식이 항상 성립한다.

$$2D_{tw}(S, Q) \geq D_{tw\_lb}(S, Q)$$

따라서 위 식과 [따름 정리 2]에 근거하여 [정리 5]는 증명된다.

2배 확장된 ε로 질의하면 특성 벡터 내에 다중 역할 요소가 존재하여도 착오 기각이 발생하지 않아 접두어 질의 기법에 L_1을 적용할 수 있다. 그러나 ε의 값이 2배가 된 것에 비례하여 착오 해답(false alarm)이 증가하게 된다. 본 논문에서는 이배수 ε 질의 시 발생되는 착오 해답을 줄여 성능을 높이기 위한 방안으로 초기 데이터 시퀀스에서 특성 벡터를 추출하여 트리를 구성할 때, 두 종류의 트리를 구성하는 방법을 제안한다. 데이터 시퀀스에서 특성 벡터를 다중 역할 요소 없이 추출할 수 있는 시퀀스로 구성된 트리 T_d와 그렇지 못한 트리 T_i로 두 종류의 트리를 구성한다. 단, 트리를 구성하는 시점에는 워핑된 시퀀스가 아니므로 다중 역할 요소의 유무를 정확히 알 수 없다. 따라서 시퀀스 내에 First값과 Last값이 최대값 혹은 최소값인 시퀀스를 모두 T_i로 구성한다. 이는 질의 시퀀스에 대해서도 동일하게 적용된다. 질의 시퀀스가 들어오면 T_d에는 정상적인 ε 질의를 시행하고 T_i에는 이배수 ε 질의를 시행한다. 만일 질의 시퀀스의 특성 벡터가 다중 역할 요소를 가지는 경우에는 T_d와 T_i 모두에 이배수 ε 질의 기법을 사용하여 질의 한다. 이 경우에는 착오 해답이 증가할 수 있다. 그러나 길이가 길고 그 값이 다양한 실제 시퀀스 데이터에서 특성 벡터를 구성하는 경우 다중 역할 요소가 존재하는 경우는 매우 드물게 나타난다.

4. 성능 평가

본 장에서는 접두어 질의 기법의 성능을 관련 연구에서 제시된 Naïve-Scan, LB-Scan, ST-Filter와 비교 분석하고자 한다. 제 4.1절에서는 성능 평가를 위한 실험 환경을 설명하고, 제 4.2절에서는 전체적인 실험 결과를 제시한다.

4.1 실험 환경

본 실험에서는 성능 분석을 위하여 실제 데이터베이스 K_Stock_Data와 합성 데이터 Syn_Data를 사용하였다. K_Stock_Data는 한국의 실제 주식 데이터로서 길이가 300인 620개의 데이터 시퀀스로 구성된다. 합성 데이터 Syn_Data 내의 각 시퀀스 $S = <s_1, s_2, ..., s_n>$는 다음과 같은 랜덤 워크(random walk) 형태를 가진다.

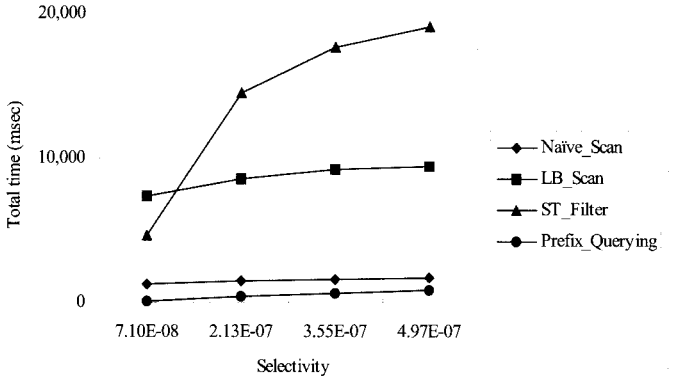
$$s_i = s_{i-1} + z_i$$

여기서 z_i는 구간 [-0.1, 0.1] 사이에서 균일한 분포를 취하는 랜덤 변수이며, 시퀀스의 첫 요소 값 s_1은 구간 [1, 10] 사이의 임의의 값을 취하도록 하였다. 확장성 실험으로 각각 1,000개, 2,000개, 3,000개, 4,000개의 길이가 200인 데이터 시퀀스들로 구성된 네 가지 Syn_Data들과 길이가 각각 200, 300, 400, 500인 1,000개의 데이터 시퀀스들로 구성된 네 가지 Syn_Data들을 생성하였다.

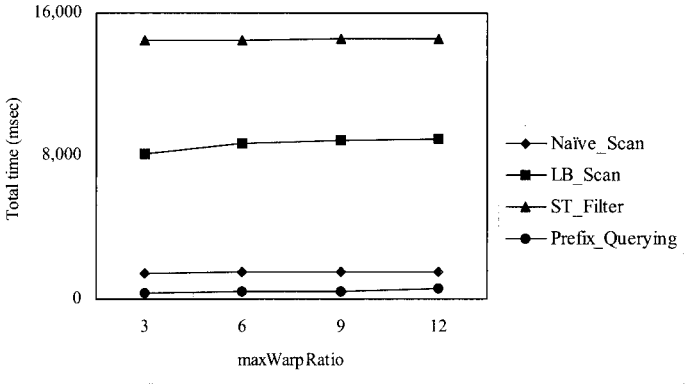
질의 시퀀스 Q는 데이터베이스에서 선택한 시퀀스로부터 길이가 Len(Q)인 임의의 서브시퀀스를 선택하여 그대로 사용하는 방법으로 생성하였다. 질의 구성 시에는 질의 선택률(query selectivity)을 아래의 식과 같이 정의하고, 각 질의에 대하여 원하는 선택률을 만족하도록 허용치 ε을 조정하였다.

$$\text{선택률} = \frac{\text{데이터베이스 내에서 질의시퀀스 Q와 } \varepsilon\text{-매치하는 모든 서브시퀀스들의 수}}{\text{데이터베이스 내에서 질의시퀀스 Q와 매치 될 수 있는 길이를 갖는 모든 서브시퀀스들의 수}}$$

성능 평가를 위한 하드웨어 플랫폼은 1.7GHz Pentium IV CPU와 1.2GB의 주기억장치가 장착된 PC이며, 소프트웨어 플랫폼은 Linux kernel version 2.4.18 및 Glibc 2.2.4이다. 실험 중 다른 프로세스들과의 상호 간섭을 방지하기 위하여 운영 체제를 단일 사용자 모드로 설정해 모든 사용자 프로세스들을 제거한 상황에서 실험하였다. 또한 ST-Filter를 위



(그림 3) 선택률 변화에 따른 성능 결과



(그림 4) maxWarpRatio 변화에 따른 성능 결과

한 도메인 분류 방법으로서 최대 엔트로피 기법(maximum entropy method)을 이용하여 ST-Filter가 50개의 구간을 갖도록 하였다.

실험의 순서는 실험에서 변화되는 인자에 대해 L_1의 Naïve-Scan, LB-Scan, ST-Filter, 접두어 질의 기법의 성능을 제시하고 분석하였다.

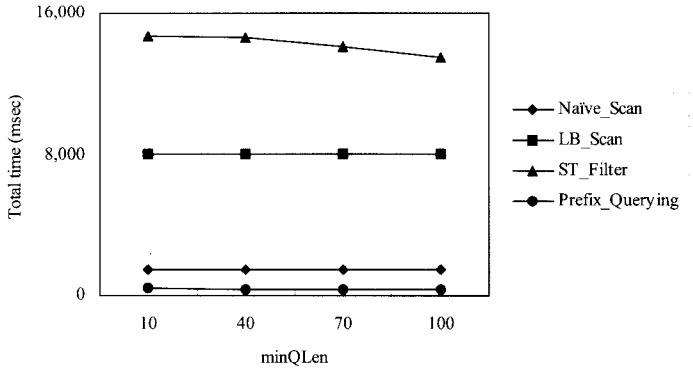
4.2 실험 결과 및 분석

실험 1에서는 선택률을 7.1×10^{-8}, 2.13×10^{-7}, 3.55×10^{-7}, 4.97×10^{-7}로 변화하면서 접두어 질의 기법과 여과 단계를 거치는 기존 기법들의 성능을 비교하였다. 사용된 질의 시퀀스의 길이는 110이며, 각각 2, 6, 10, 14개를 최종 결과로 반환하였다. (그림 3)은 실험 결과를 나타낸 것이다.

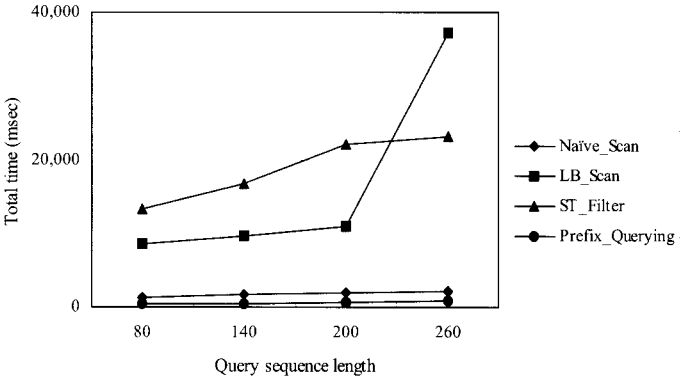
선택률이 증가함에 따라 모든 기법에서 검색 시간이 증가하였다. ST-Filter 기법의 경우 그 증가폭이 크게 나타나고 있다. 이는 L_1의 경우 접미어 트리에서 후보를 검색할 때 거리 함수 테이블을 구성하는데 요구되는 CPU 연산 양이 ε 값에 많은 영향을 받기 때문이다. LB-Scan 기법의 경우 그 특성상 후보 시퀀스를 구성 할 때 모든 데이터 시퀀스를 액세스해야 하므로 비교적 좋지 않는 성능을 나타내고 있다.

본 논문의 실험에서 주목할 점은 모든 경우에서 Naïve-Scan 기법이 LB-Scan나 ST-Filter 기법 보다 성능이 우수한 것으로 나타난 것이다. 이것은 본 논문의 실험에서 CPU 처리 과정을 최적화 한 개선된 Naïve-Scan[23]을 사용하였기 때문이다. 본 논문에서 사용된 Naïve-Scan 방식은 질의 시퀀스와 서브시퀀스들 간의 타임 워핑 거리를 계산하는 과정에서 발생되는 많은 중복 작업을 사전에 제거하는 방식으로 CPU 성능을 극대화하는 방식이다. 따라서 이 Naïve-Scan 방식을 이용하면 기존에 CPU 처리에서 발생하는 성능 병목 현상이 줄어들어 본 실험과 같이 우수한 성능을 보이는 것이다. 이러한 성능 역전 현상은 이후의 모든 실험에서 일관되게 나타났으며, 이는 참고 문헌 [23]의 실험 결과와도 부합되는 것이다. 성능 역전 현상에 대해서는 이후의 실험에서 별도로 언급하지 않는다. 접두어 질의 기법은 다른 기법들과 비교하여 월등한 성능을 보였으며, 기존 기법 중 가장 좋은 성능을 보인 Naïve-Scan에 비해 최대 10.7배 더 좋은 성능을 보였다. 이것은 인덱스를 이용함으로써 여과 단계의 성능을 크게 개선할 수 있기 때문이다.

시계열 데이터에 타임 워핑 기법을 적용 할 때, 경우에 따라 하나의 시퀀스 요소가 너무 많이 반복되지 않도록 한 요소가 반복될 수 있는 최대의 수를 정해 놓는다. 이렇게 정해놓은 수를 maxWarpRatio라고 한다[15]. [실험 2]에서는 maxWarpRatio를 3, 6, 9, 12로 변화시키면서 접두어 질의 기법과 기존 기법들의 성능을 비교하였다. 사용된 질의 시퀀스의 길이는 110이고 선택률은 2.13×10^{-7}이다. (그림 4)는 실험 결과를 나타낸 것이다.



(그림 5) minQLen 변화에 따른 성능 결과



(그림 6) 질의 시퀀스 길이 변화에 따른 성능 결과

모든 기법에서 maxWarpRatio가 증가함에 따라 여과 단계에서 결과로 반환되는 서브시퀀스들의 수와 후보 서브시퀀스의 수가 많아져서 소요 시간이 조금씩 증가하는 것을 볼 수 있다. 접두어 질의 기법의 경우 최소 질의 시퀀스 길이(minQLen)[19]를 maxWarpRatio로 나눈 값을 윈도우 크기 $w = \lceil \frac{\min QLen}{\max WarpRatio} \rceil$로 사용하므로 maxWarpRatio가 커지면 접두어의 수가 증가하고 이로 인해 검색된 후보 서브시퀀스의 수가 증가하고 있어 후처리 시간이 점차 증가하는 것으로 나타났다. 이 실험에서 접두어 질의 기법은 나머지 기법 중 가장 좋은 성능을 보인 Naïve-Scan 기법에 비해 최대 4.4배 더 좋은 성능을 보이고 있다.

실험 3에서는 minQLen를 10, 40, 70, 100으로 변화시키면서 접두어 질의 기법과 기존 기법들의 성능을 비교하였다. 사용된 질의 시퀀스의 길이는 110이고, 선택률은 2.13×10^{-7}이다. (그림 5)은 실험 결과를 나타낸 것이다.

ST-Filter 기법은 minQLen가 증가할수록 접미어 트리에 포함되는 접미어의 수가 감소하므로 여과 과정의 계산량이 줄어들어 전체 검색 시간이 줄어드는 것으로 나타났다. 다른 기법들에서는 minQLen의 값에 거의 영향을 받지 않고 대부분 일정한 검색 성능을 보였다. 이 실험에서도 역시 접두어 질의 기법은 나머지 기법 중 가장 좋은 성능을 보인 Naïve-Scan 기법에 비해 최대 3.8배 더 좋은 성능을 보이고 있다.

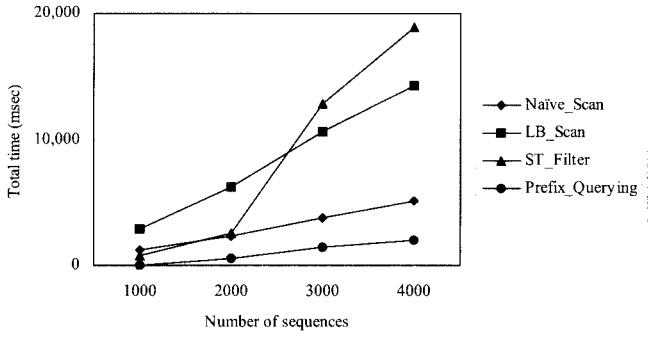
실험 4에서는 질의 시퀀스의 길이를 80, 140, 200, 260으로 변화시키면서 접두어 질의 기법과 기존 기법들의 성능을 비교하였다. 사용된 질의의 선택률은 2.13×10^{-7}이다. (그림 6)은 실험 결과를 나타낸 것이다.

질의 시퀀스의 길이가 증가함에 따라 거리 계산 비용이 증가하게 되고, 이로 인해 여과 단계와 후처리 단계에서 소요되는 시간은 모두 그만큼 증가하게 된다. ST-Filter 기법과 LB-Scan 기법은 매우 급격한 검색 시간의 증가를 보여준다. 반면, Naïve-Scan과 접두어 질의 기법은 질의 시퀀스 증가에 따라 매우 완만한 검색 시간의 증가를 보였다. 이 실험에서 역시 접두어 질의 기법은 가장 우수한 성능을 보였으며, Naïve-Scan과 비교하여 최대 3.6배의 성능 개선을 나타냈다.

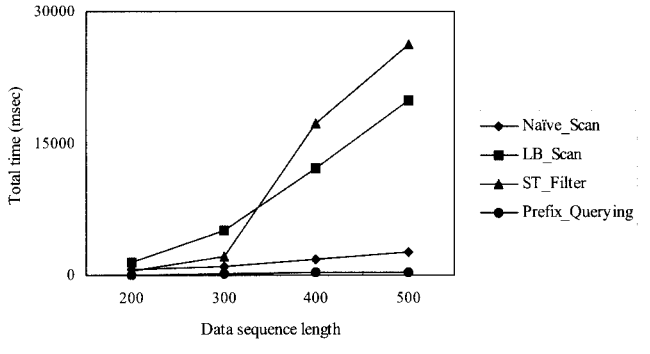
이전의 실험들에서 사용한 주식 데이터는 비교적 소규모의 데이터베이스이다. 이후의 실험에서는 조금 더 대규모 데이터베이스에서 성능 평가를 수행하기 위하여 다양한 시퀀스 길이와 수를 가지는 생성 데이터를 이용해서 실험하였다.

실험 5에서는 데이터 시퀀스의 길이는 200으로 고정시킨 상태에서 데이터 시퀀스들의 수를 1,000, 2,000, 3,000, 4,000으로 변화하면서 접두어 질의 기법과 기존 기법의 성능을 비교하였다. 사용된 질의 시퀀스의 길이는 60이며, 질의 선택률은 1.47×10^{-7}이다. (그림 7)는 실험 결과를 나타낸 것이다.

시퀀스 개수가 증가함에 따라 모든 기법에서 검색 시간이



(그림 7) 데이터 시퀀스 개수 변화에 따른 성능 결과



(그림 8) 데이터 시퀀스 길이 변화에 따른 성능 결과

그와 비례하여 증가하고 있음을 알 수 있다. ST-Filter와 LB-Scan은 매우 급격한 성능 저하를 보이는 반면, Naïve-Scan과 접두어 질의 기법은 상대적으로 완만한 성능 저하를 보였다. 이 실험에서도 역시 접두어 질의 기법의 성능이 가장 우수하게 나타났으며, 기존의 가장 나은 기법과 비교하여 최대 180배까지의 성능 개선 효과를 보였다.

실험 6에서는 데이터 시퀀스의 개수는 1,000으로 고정시킨 상태에서 데이터 시퀀스들의 길이를 200, 300, 400, 500으로 변화하면서 접두어 질의 기법과 기존 기법의 성능을 비교하였다. 사용된 질의 시퀀스의 질의 선택률은 1.47×10^{-7}이다. (그림 8)은 실험 결과를 나타낸 것이다.

시퀀스의 길이가 증가함에 따라 모든 기법들의 검색 시간이 증가하고 있다. 특히 ST-Filter와 LB-Scan의 경우 실험 5에서와 매우 비슷한 결과를 보여 주고 있다. 이 실험에서 LB-Scan은 여과 단계에서 모든 시퀀스 요소들에 대하여 검색을 하여야 하므로 시퀀스의 길이가 길어지는 경우에도 실험 5와 일치하는 결과를 보였다. 이 실험에서도 역시 접두어 질의 기법은 Naïve-Scan의 성능과 비교하여 최대 108배 개선된 가장 뛰어난 성능을 보였다.

5. 결 론

접두어 질의 기법은 착오 기각 없이 타임 워핑 하의 시계열 서브시퀀스 매칭을 처리하는 인덱스를 이용한 최초의 방식이다. 이 기법은 사용자가 질의를 편리하게 작성하도록 하기 위하여 기본 거리 함수로서 L_∞를 사용한다. 본 논문에서는 L_∞ 대신 타임 워핑 하의 시계열 서브시퀀스 매칭에서 기본 거리 함수로서 가장 널리 사용되는 L_1을 적용할 수 있도록 접두어 질의를 확장하는 방안에 대하여 논의하였다. 본 논문의 주요 공헌은 아래와 같이 요약될 수 있다.

- L_∞ 대신 L_1을 기본 거리 함수로서 적용할 수 있도록 접두어 질의 기법을 확장하였다.
- 확장된 접두어 질의 기법을 이용한 타임 워핑 하의 시계열 서브시퀀스 매칭에서 착오 기각이 발생하지 않음을 이론적으로 증명하였다.
- 다양한 실험에 의한 기존 기법들과의 성능 비교를 통하여 확장된 접두어 질의 기법의 우수성을 규명하였다. 실험 결과에 의하면, 확장된 접두어 질의 기법은 기존의 가장 좋은 성능을 보이는 기법과 비교하여 매우 극적인 성능 개선 효과를 보이는 것으로 나타났다.

제4장의 실험 5에서와 같이, 대용량의 데이터베이스의 경우 확장된 접두어 질의 기법에서도 여과 단계 후에 발생되는 후보 서브시퀀스들의 개수가 증가되어 성능이 저하된다. 따라서 여과 단계에서 후보 서브시퀀스들의 수를 좀더 줄일 수 있는 방안이 요구된다. 이를 위하여 향후 연구로서 타임 워핑 거리에 좀더 가까운 값을 반환하는 하한 함수를 고안하는 것을 고려하고 있다.

참 고 문 헌

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

[2] C. Chatfield, *The Analysis of Time-Series : An Introduction*, Third Edition, Chapman and Hall, 1984.

[3] R. Agrawal et al., "Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time- Series Databases," In *Proc. Int'l. Conf. on Very Large Data Bases*, VLDB, pp.490-501, Sept., 1995.

[4] 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.

[5] M. S., Chen, J., Han, and P. S., Yu, "Data Mining : An Overview from Database Perspective," *IEEE Trans. on Knowledge and Data Engineering*, Vol.8, No.6, pp. 866-883, 1996.

[6] D. Rafiei and A. Mendelzon, "Similarity-Based Queries for Time-Series Data," In *Proc. Int'l. Conf. on Management of Data*, ACM SIGMOD, pp.13-24, 1997.

[7] B. K. Yi and C. Faloutsos, "Fast Time Sequence Indexing for Arbitrary L_p Norms," In *Proc. Int'l. Conf. on Very Large Data Bases*, VLDB, pp.385-394, 2000.

[8] K. P. Chan and A. W. C. Fu, "Efficient Time-Series Matching by Wavelets," In *Proc. Int'l. Conf. on Data Engineering*, IEEE ICDE, pp.126-133, 1999.

[9] K. K. W. Chu and M. H. Wong, "Fast Time-Series Searching with Scaling and Shifting," In *Proc. Int'l. Symp. on Principles of Database Systems*, ACM PODS, pp.237-248, May, 1999.

[10] D. Q. Goldin and P. C. Kanellakis, "On Similarity Queries for Time-Series Data : Constraint Specification and Implementation," In *Proc. Int'l. Conf. on Principles and Practice of Constraint Programming*, CP, pp.137-153, Sept., 1995.

[11] D. Rafiei, "On Similarity-Based Queries for Time Series Data," In *Proc. Int'l. Conf. on Data Engineering*, IEEE ICDE, pp.410-417, 1999.

[12] G. Das, D. Gunopulos, and H. Mannila, "Finding Similar Time Series," In *Proc. European Symp. on Principles of Data Mining and Knowledge Discovery*, PKDD, pp.88-100, 1997.

[13] W. K. Loh, S. W. Kim, and K. Y. Whang, "Index Interpolation : An Approach for Subsequence Matching Supporting Normalization Transform in Time- Series Databases," In *Proc. ACM Int'l. Conf. on Information and Knowledge Management*, ACM CIKM, pp.480-487, 2000.

[14] W. K. Loh, S. W. Kim, and K. Y. Whang, "Index Interpolation : A Subsequence Matching Algorithm Supporting Moving Average Transform of Arbitrary Order in Time-Series Databases," *IEICE Trans. on Information and Systems*, Vol.E84-D, No.1, pp.76-86, 2001.

[15] 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.

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

[17] 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 ICDE, pp.23-32, 2000.

[18] 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. Int'l. Conf. on Data Engineering*, IEEE ICDE, pp.607-614, 2001.

[19] S. H. Park, S. W. Kim, J. S. Cho, and S. Padmanabhan, "Prefix-Querying : An Approach for Effective Subsequence Matching Under Time Warping in Sequence Databases," In *Proc. ACM Intl. Conf. on Information and Knowledge Management*, ACM CIKM, pp.255- 262, 2001.

[20] L. Rabiner and H. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[21] Sang-Wook Kim, Sang-Hyun Park, and Wesley W. Chu, "Efficient Processing of Similarity Search Under Time-Warping in Sequence Databases : An Index- Based Approach," *Information Systems*, Vol.29, No.5, pp.405-420, Jul., 2004.

[22] C. Faloutsos, *private communication*, 2001.

[23] Man-Soon Kim, Sang-Wook Kim, and Mi-Young Shin, "Optimization of Subsequence Matching Under Time Warping in Time-Series Databases," *ACM Symp. on Applied Computing*, pp.581-586, Apr., 2005.

[24] G. A. Stephen, *String Searching Algorithms*, World Scientific Publishing, 1994.

[25] E. Keogh, "Exact Indexing of Dynamic Time Warping", In *Proc. Int'l. of the 28th VLDB Conference*, 2002.

장 병 철

e-mail : bcchang@ihanyang.ac.kr

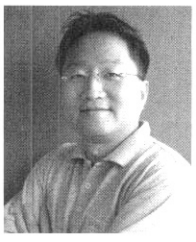
1996년 안동대학교 컴퓨터공학과(학사)

2001년 한양대학교 컴퓨터교육학과(석사)

2003년~현재 한양대학교 정보통신학과 박사과정

관심분야 : 데이터마이닝, 시맨틱웹, e-Learning

김 상 욱



e-mail : wook@hanyang.ac.kr
1989년 서울대학교 컴퓨터공학과(학사)
1991년 한국과학기술원 전산학과(석사)
1994년 한국과학기술원 전산학과(박사)
1991년 미국 Stanford University, Computer Science Dapartment 방문연구원
1994년~1995년 KAIST 정보전자연구소 전문연구원
1999년~2000년 미국 IBM T.J Watson Research Center Post-Doc.
1995년~2000년 강원대학교 컴퓨터정보통신학부 부교수
2003년~현재 한양대학교 정보통신대학 정보통신학부 교수
관심분야 : 데이터베이스 시스템, 저장시스템, 데이터 마이닝, 바이오 정보공학, 멀티미디어정보검색, 공간 데이터베이스/GIS 주기억장치데이터베이스, 트랜잭션 관리

차 재 혁

e-mail : chajh@hanyang.ac.kr
1987년 서울대학교 계산통계학과(학사)
1991년 서울대학교 컴퓨터공학과(석사)
1997년 서울대학교 컴퓨터공학과(박사)
1997년~1998년 한국학술진흥재단부설 첨단학술정보센터 선임연구원
1998년~2001년 한양대학교 사범대학 컴퓨터교육과 조교수
2001년~현재 한양대학교 정보통신대학 정보통신학부 부교수
관심분야 : XML 데이터베이스, 플래시메모리 기반 저장시스템, 멀티미디어 콘텐츠적응화, e-Learning