타임 워핑 하의 시계열 서브시퀀스 매칭 기법의 성능 평가

Performance Evaluation of Methods for Time-Series Subsequence Matching Under Time Warping

김만순, 김상욱*
강원대학교, 한양대학교*

Kim Man-Soon, Kim Sang-Wook*
Kangwon National University, Hanyang University*

요약

시계열 데이터베이스란 객체의 변화되는 값들의 연속으로 구성된 데이터 시퀀스들의 집합이며, 타임 워핑 하의 서브시퀀스 매칭은 주어진 질의 시퀀스와 타임 워핑 거리가 허용치 이하인 서브시퀀스들을 시계열 데이터베이스로부터 찾아내는 연산이다. 본 논문에서는 먼저 타임 워핑 하의 시퀀스 매칭을 지원하는 기존의 기법들의 특성을 지적하고, 이들을 전체 매칭 및 서브시퀀스 매칭에 각각 적용하는 방안에 관하여 논의한다. 또한, 실제 주식 데이터를 이용한 다양한 실험을 통하여 이들에 대한 정량적인 성능 평가를 수행한다. 타임 워핑 하의 서브시퀀스 매칭을 위한 기존 기법들의 성능을 상호 비교한 연구 결과는 아직 제시된 바 없다. 따라서 본 연구 결과는 이러한 세 가지 기법들에 대한 성능을 제시하는 좋은 자료로서 사용될 수 있을 것이다.

Abstract

A time-series database is a set of data sequences, each of which is a list of changing values corresponding to an object. Subsequence matching under time warping is defined as an operation that finds such subsequences whose time warping distance to a given query sequence are below a tolerance from a time-series database. In this paper, we first point out the characteristics of the previous methods for time-series sequence matching under time warping, and then discuss the approaches for applying them to whole matching as well as subsequence matching. Also, we perform quantitative performance evaluation via a series of experiments with real-life data. There have not been such researches in the literature that compare the performances of all the previous methods of subsequence matching under time warping. Thus, our results would be used as a good reference for showing their relative performances.

I. 서론

시계열 데이터베이스(time-series database)란 객체의 변화되는 값들의 연속으로 구성된 데이터 시퀀스(data sequence)들의 집합이다[Agr93]. 대표적인 예로는 주가 데이터, 환율 데이터, 기온 데이터 등이 있다[Cha84][Agr95][Fal94]. 시퀀스 매칭(sequence matching)이란 시퀀스 S1, S2, ..., SN을 포함하는 시계열 데이터베이스 D로부터 질의 시퀀스(query sequence) Q와 유사한 시퀀스(전체 매칭이라 함) 혹은 서브시퀀스(서브시퀀스 매칭이라 함)들을 검색하는 연산이다. 시퀀스 매칭은 데이터 마이닝(data mining) 및 데이터 웨어하우징(data warehousing) 분야의 중요한 연산으로 사용된다[Agr93][Agr95][Fal94] [Che96][Raf97].

시퀀스 매칭에 관한 기존의 많은 연구에서는 길이 n의 시퀀스를 n 차원 공간상의 한 점으로 간주한다. 또한, 길이 n인 서로 다른 두 시퀀스 X(=[x1, x2, ... , xn])와 Y(=[y1, y2, ..., yn])간의 유사한 정도를 측정하는 척도로서 아래의 식과 같이 정의되는 거리 함수 Lp(X,Y)를 널리 사용한다.

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

L1은 맨하탄 거리(Manhattan distance), L2는 유클리드 거리(Euclidean distance), L∞은 대응되는 각 요소 값 쌍의 거리 중 최대 거리를 의미한다[Yi00]. 응용에서 주어진 허용치 ε보다 작은 Lp(X,Y)를 갖는 임의의 두 시퀀스 X, Y를 유사하다고 간주한다[Agr93] [Cha99] [Chu99] [Fal94] [Gol95] [Raf97] [Raf99].

Lp 거리 함수만을 이용한 시퀀스 매칭을 통해서는 사용자가 원하는 시퀀스들을 검색하지 못하는 경우가 빈번하게 발생한다. 타임 워핑은 시퀀스내의 각 요소 값을 임의의 수만큼 반복시키는 것을 허용하는 변환이다[Yi98]. 타임 워핑 후의 두 시퀀스들 간의 거리를 타임 워핑 거리(time warping distance)라 한다. 타임 워핑 거리는 데이터베이스내의 시퀀스들의 길이가 서로 달라서 Lp 거리 함수를 이용하여 유사 정도를 직접 측정할 수 없는 경우에 매우 유용하다[Kim01]. 타임 워핑 하의 서브시퀀스 매칭이란 질의 시퀀스와의 타임 워핑 거리가 주어진 허용치 ε보다 작은 서브시퀀스들을 시계열 데이터베이스로부터 찾는 연산으로 정의된다.

최근, 타임 워핑 하의 시퀀스 매칭에 관한 다양한 연구가 수행되어 왔다[Ber96][Yi98][Par00][Kim01] [Par01]. 타임 워핑 하의 시퀀스 매칭의 처리를 위한 기존의 기법들은 크게 여과 단계(filtering step)와 후처리 단계(post processing step)로 구성된다. 여과 단계는 주어진 질의 시퀀스와 전혀 유사하지 않는 시퀀스들을 미리 제거함으로써 최종 결과에 포함될 가능성이 매우 높은 시퀀스들로 구성되는 후보 집합(candidate set)을 구성하는 단계이다. 질의 시퀀스와 실제로 유사한 시퀀스를 여과 단계에서 후보 집합 내에 포함시키지 못하는 현상을 착오 기각(false dismissal)이라 한다. 반면, 질의 시퀀스와 유사하지 않은 일부의 시퀀스를 여과 단계에서 후보 집합 내에 포함시키는 현상을 착오 채택(false alarm)이라 한다. 후처리 단계는 후보 집합에 속하는 각 시퀀스를 디스크로부터 액세스하여 이것이 질의 시퀀스와 유사한가의 여부를 판단함으로써 착오 채택을 제거하는 단계이다.

타임 워핑 하의 시퀀스 매칭을 위한 기존의 기법들은 Naive-Scan[Ber96], LB-Scan[Yi98], ST-Filter[Par00] 등이 있다[1]. Naive-Scan과 LB-Scan은 각각 타임 워핑 하의 전체 매칭을 위하여 제안된 기법이고, ST-Filter는 타임 워핑 하의 서브시퀀스 매칭을 위하여 제안된 기법이다.

본 논문에서는 먼저 이 방법들을 기반으로 전체 매칭 및 서브시퀀스 매칭의 수행을 위한 확장 방안에 관하여 논의하고, 각 기법의 장단점을 지적한다. 또한, 실제 주식 데이터를 이용한 다양한 실험을 통하여 이들에 대한 정량적인 성능 평가를 수행한다. 타임 워핑 하의 서브시퀀스 매칭을 위한 이 세 기법들 간의 정량적인 성능 분석에 관한 연구 결과는 아직 제시된 바 없다. 따라서 본 연구 결과는 세 가지 기법들에 대한 성능을 비교할 수 있는 좋은 자료로서 사용될 수 있을 것이다.

본 논문의 구성은 다음과 같다. 제 2장에서는 논리 전개를 위한 용어를 정의한다. 제 3장에서는 관련 연구로서 타임 워핑 하의 시퀀스 매칭을 위한 기존의 기법들을 소개하고, 장단점을 논의한다. 제 4장에서는 다양한 실험들을 통하여 기존 기법들의 성능을 평가한다. 끝으로, 제 5장에서는 본 논문을 요약하고, 결론을 내린다.

II. 용어 정의

두 시퀀스 S와 Q간의 타임 워핑 거리(time warping distance) Dtw는 다음과 같이 재귀적으로 정의된다[Yi98][Par00][Kim01]:

정의 1:

(1) Dtw((), ()) = 0,

1) 이러한 세 가지 기법들 이외에도 참고 문헌 [Yi98]에서 제안한 FastMap 기반 기법, 참고 문헌 [Kim01][Par01]에서 제안한 인덱스 기반 기법(index-based approach) 등이 있다. FastMap 기반 기법은 다차원 인덱스(multidimensional index)를 이용한 여과 단계를 통하여 빠른 처리 성능을 제공하나, 착오 기각을 유발시키는 문제점이 있다. 또한, 인덱스 기반 기법은 착오 기각을 유발시키지 않으면서도 다차원 인덱스를 이용한 여과 단계를 수행하는 좋은 기법이나, 타임 워핑 거리 계산을 위하여 L1을 기본 거리 함수로 사용하는 위의 세 가지 기법들과는 달리, L∞을 기본 거리 함수로 사용한다[Kim01][Par01]. 즉, FastMap 기반 기법과 인덱스 기반 기법은 위의 세 기법들과는 적용 응용 분야 측면에서 차이가 있다. 본 논문에서는 이러한 이유로 인하여 이 두 가지 기법들을 이후의 논의의 대상에서 제외한다.

(2) Dtw(S, ()) = Dtw((), Q) = ∞,

(3) Dtw(S, Q) = (|Lp(First(S), First(Q))|p + |min(Dtw(S, Rest(Q)), Dtw(Rest(S), Q), Dtw(Rest(S), Rest(Q)))|p)1/p

여기서, First(S)는 각각 S의 첫 번째 요소 s1을 의미하며, Rest(S)는 s1을 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. Lp는 응용에서 적합한 것을 선택하여 사용할 수 있다. 본 논문에서는 현재 가장 널리 사용되는 맨해튼 거리(Manhattan distnace) L1을 기반으로 하는 타임 워핑 거리에 연구의 초점을 맞추고자 한다. □

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>를 타임 워핑에 의하여 동일한 시퀀스 <20, 20, 21, 21, 20, 20, 23, 23, 23>으로 변환될 수 있으며, 이 결과 Dtw(S, Q)는 0이 된다.

전술한 바와 같이, Lp 거리 함수는 시퀀스들의 길이가 동일한 경우에만 유사한 정도를 측정할 수 있다. 반면, 타임 워핑 거리는 데이터베이스내의 시퀀스들의 길이가 서로 달라서 Lp 거리 함수를 이용하여 유사 정도를 직접 측정할 수 없는 경우에 매우 유용하다[Kim01]. 현재, 타임 워핑은 음성 인식 분야에서 널리 사용되고 있으며[Rab93], 심전도 데이터, 주가 데이터, 기온 데이터, 기업 성장률 데이터 등에도 동일한 방식으로 적용할 수 있다.

III. 기존의 기법들에 관한 고찰

본 장에서는 관련 연구로서 타임 워핑 하의 시퀀스 매칭을 착오 기각 없이 수행하는 기존의 연구들로서 Naive-Scan, LB-Scan, ST-Filter를 소개한다. 각 기법에 관하여 (1) 전체 매칭 방안, (2) 서브시퀀스 매칭 방안[2], (3) 주요 특징에 관하여 논의한다.

1. Naive-Scan[Ber96]

전체 매칭 방안 : 디스크로부터 각 데이터 시퀀스를 액세스한 후, 이 데이터 시퀀스 S와 질의 시퀀스 Q의 타임 워핑 거리 Dtw(S, Q)를 계산함으로써 전체 매칭을 수행한다. Dtw(S, Q)를 효과적으로 계산하기 위한 방법으로서 동적 프로그래밍(dynamic programming)을 사용한다[Ber96]. 계산된 Dtw(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)는 다음과 같은 재귀 관계(recurrence relation)에 의하여 구성된다[Ber96]. 동적 프로그래밍 알고리즘은 위의 재귀 관계를 이용하여 거리 축적 테이블 T를 아래에서 위로 채워 나간다.

T(0,0) = 0

T(0,j) = T(i,0) = ∞

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

다음의 그림 2.1은 기본 거리 함수로서 L1이 사용되는 경우, 거리 축적 테이블을 이용한 두 시퀀스 S와 Q의 타임 워핑 거리 계산 예를 보인다. 계산 결과, Dtw(S, Q)는 12가 된다.

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

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 2.1 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 Dtw(S[i:j], Q)를 동적 프로그래밍을 이용하여 계산함으로써 서브시퀀스 매칭을 수행한다.

특징 : 여과 단계를 거치지 않으므로 후처리 단계의 수행 시간이 지나치게 크다[Kim01]. 즉, 모든 데이터 시퀀스들을 디스크로부터 액세스해야 한다는 부담이 있다. 또한, (서브)시퀀스 S와 Q의 Dtw를 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.

2. LB-Scan[Yi98]

전체 매칭 방안 : 타임 워핑 거리 Dtw의 반환 값보다 항상 작은 값을 반환하는 하한 함수(lower-bound function) Dlb를 이용하여 여과 단계를 수행한다. 즉, 여과 단계에서는 디스크로부터 각 데이터 시퀀스를 액세스한 후, 이 데이터 시퀀스 S와 질의 시퀀스 Q에 대하여 Dlb(S, Q)를 적용한다. 여과 단계에서 Dlb의 반환 값이 허용치 ε이하인 데이터 시퀀스 S'에 대해서는 질의 시퀀스 Q와의 타임 워핑 거리 Dtw(S', Q)를 계산하는 후처리 단계를 수행한다. Dtw(S', Q)의 계산을 위하여 Naive-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 Dlb(S[i:j], Q)를 적용한다. 여기서에서 Dlb의 반환 값이 허용치 ε이하인 서브시퀀스 S[i:j]'에 대해서는 동적 프로그래밍을 이용하여 질의 시퀀스 Q와의 타임 워핑 거리 Dtw(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 Naive-Scan과 동일하다. 여과 단계에서는 모든 (서브)시퀀스 S와 질의 시퀀스 Q 간의 Dlb가 계산된다. 각 Dlb(S, Q)를 계산할 때의 CPU 수행 시간은 O(|S|+|Q|)로서 O(|S|*|Q|)인 Dtw(S, Q)와 비교하여 CPU 수행 시간이 매우 작다[Yi00]. 여과 단계를 통하여 최종 결과에 포함될 가능성이 없는 (서브)시퀀스들을 사전에 제외시킬 수 있으므로 후처리 단계의 수행 시간을 크게 줄일 수 있다[Kim01]. 따라서 여과 단계에서 제외되는 (서브)시퀀스들이 많은 경우, 성능 개선 효과는 매우 크다.

3. ST-Filter[Par00]

전체 매칭 방안 : 여과 단계를 위하여 데이터베이스 내의 각 시퀀스의 요소 값들을 심볼로 변환시키고, 이들을 접미어 트리(suffix tree)[Ste94] 내에 저장시킨다. 여과 단계에서는 접미어 트리 검색을 이용하여 질의 시퀀스 Q와의 타임 워핑 거리 Dtw가 허용치 ε이하일 가능성이 있는 후보 시퀀스 S'들을 걸러낸다. 후처리 단계에서는 이러한 S들을 대상으로 동적 프로그래밍을 사용하여 Dtw(S', Q)를 계산한다.

서브시퀀스 매칭 방안 : 여과 단계를 위하여 각 데이터 시퀀스내 각 접미어의 요소 값들을 심볼로 변환시키고, 이들을 접미어 트리 내에 저장시킨다. 여과 단계에서는 접미어 트리 검색을 통하여 질의 시퀀스 Q와의 타임 워핑 거리 Dtw가 허용치 ε이하일 가능성이 있는 후보 서브시퀀스 S[i:j]'들을 걸러낸다. 후처리 단계에서는 이러한 S[i:j]'들을 대상으로 Dtw(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 시퀀

스들이 저장된 파일보다 훨씬 큰 것이 일반적이다. 또한, 좋은 시퀀스 매칭 성능을 제공하기 위한 최적의 도메인 분류(categorization) [Par00]가 쉽지 않으며 [Kim01], 동일한 데이터베이스의 경우에도 질의 시퀀스마다 서브시퀀스 처리 성능이 크게 다르다는 것이 문제점으로 지적된다.

Ⅳ. 성능 평가

본 장에서는 Naive_Scan, LB_Scan, ST_Filter의 성능을 비교 분석하고자 한다. 제 4.1절에서는 성능 평가를 위한 실험 환경을 설명하고, 제 4.2절에서는 실험 결과를 제시하고 분석한다.

1. 실험 환경

본 연구에서는 성능 분석을 위하여 한국의 실제 주식 데이터로서 길이가 300인 620개의 데이터 시퀀스들로 구성된 K_Stock_Data를 사용하였다.

질의 시퀀스 Q는 데이터베이스로부터 임의로 선택한 한 시퀀스로부터 길이가 Len(Q)인 임의의 위치의 서브시퀀스를 선택하여 "그대로" 사용하는 방법으로 생성하였다. 질의 구성 시에는 참고 문헌 [Moo01]에 나타난 바와 같이, 질의 선택률(query selectivity)[Moo01]을 정의하고, 각 질의에 대하여 원하는 선택률을 만족하도록 허용치 ε을 조정하였다. 또한, 성능 지수로는 서브시퀀스 매칭의 수행 시간을 사용하였으며, 동일한 환경에서 서로 다른 질의 시퀀스 50개에 대한 수행 시간을 측정하여 그 평균값을 구하였다.

실험을 위한 하드웨어 플랫폼은 1.7 GHz Pentium IV와 1,280 MB의 주기억장치가 장착된 PC를 사용하였으며, 소프트웨어 플랫폼은 운영 체제 Linux Kernel Version 2.4.18 및 컴파일러 Glibc 2.2.4를 사용하였다. 실험 중 다른 시스템 및 사용자 프로세스들과의 상호 간섭을 방지하기 위하여 운영 체제를 단일 사용자 모드로 설정하여 모든 사용자 프로세스들을 제거한 상황에서 실험하였다. 또한, 버퍼링 효과를 피하고, 실제 디스크 액세스를 보장하도록 하기 위하여 버퍼를 이용하지 않는 I/O 시스템 콜(system call)을 사용하였다.

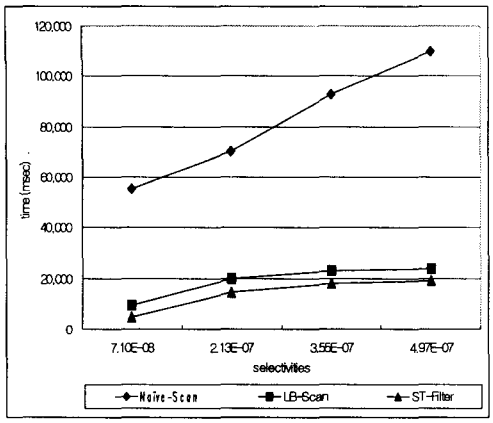
ST_Filter를 위한 도메인 분류(domain categorization) 방법으로서 최대 엔트로피 기법(maximum entropy method)을 이용하여 도메인이 50개의 구간을 갖도록 하였다.

2. 실험 결과 및 분석

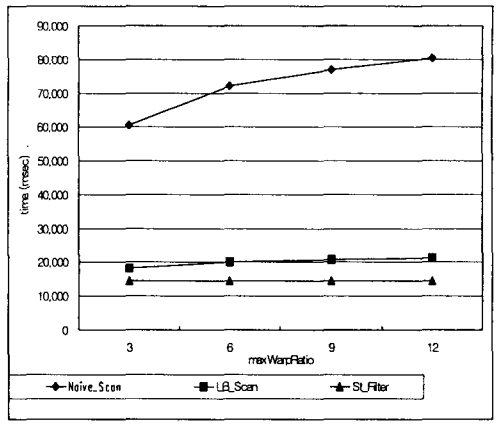
실험 1에서는 선택률을 다양하게 변화시키면서 Naive-Scan, LB-Scan, ST-Filter 등 기존 기법들의 성능의 변화를 관찰하였다. 사용된 질의 선택률은 7.1×10-8(최종 결과: 2개), 2.13×10-7(최종 결과: 6개), 3.55×10-7(최종 결과: 10개), 4.97×10-7(최종 결과: 14개)이다. 질의 시퀀스의 길이는 110로 설정하였으며, maxWarpRatio는 5로 설정하였다. 여기서, maxWarpRatio는 타임 워핑시 시퀀스 내의 각 요소 값이 최대로 반복할 수 있는 회수이다[Ber96].

그림 4.1은 선택률의 변화에 따르는 각 기법의 수행 시간의 변화를 그래프로 표현한 것이다. 가로축은 선택률의 크기를 나타내며, 세로축은 타임 워핑 하의 서브시퀀스 매칭을 처리할 때 소요되는 전체 시간을 나타낸다.

먼저, Naive-Scan의 성능의 특성을 살펴보자. Naive-Scan은 선택률이 증가함에 따라 전체 처리 시간이 급격히 증가하는 것으로 나타났다. 그 이유는 다음과 같이 요약된다. 질의 시퀀스 q와 전혀 유사하지 않아 최종 결과에 포함될 수 없는 서브시퀀스 s가 있다고 하자. 동일한 질의 시퀀스에 대하여 서브시퀀스 매칭의 선택률과 허용치 ε은 비례한다. 따라서 작은 선택률을 갖는 서브시퀀스 매칭의 허용치 ε은 작으며, 이 결과, s와 q간의 타임 워핑 거리를 계산하기 위하여 생성하는 거리 축적 테이블의 몇 개의 행들만을 채움으로써 해당 s가 q와 유사하지 않다는 것을 바로 판단할 수 있는 것이다. 반대로, 큰 선택률을 갖는 서브시퀀스 매칭에서는 거리 축적 테이블의 많은 수의 행들을 채운 후에야 해당 s가 q와 유사하지 않다는 것을 파악할 수 있으므로 상대적으로 처리 시간이 급격히 증가하는 것이다.



▶▶ 그림 4.1 선택률 변화에 따르는 전체 수행 시간



▶▶ 그림 4.2 maxWarpRatio 변화에 따르는 전체 처리 시간

LB-Scan과 ST-Filter의 경우, 선택률이 증가함에 따라 전체 처리 시간이 증가하는 경향을 보였으나, 증가 정도는 Naive-Scan과 비교하여 작은 것으로 나타났다. LB-Scan과 ST-Filter는 Naive-Scan과는 달리 여과 단계를 거치므로 선택률이 증가함에 따라 많은 CPU 처리 시간을 요구하는 거리 축적 테이블의 구성을 미리 회피할 수 있기 때문이다. 모든 선택률의 범위에서 Naive-Scan이 가장 떨어지는 성능을 나타내었다. 이것은 여과 단계를 거치지 않기 때문이다. 또한, 여과 단계를 사용하는 LB-Scan과 ST-Filter 중 ST-Filter가 더 좋은 성능을 나타냈었다. 이는 여과 단계에서 인덱스를 사용하지 않는 LB-Scan과는 달리, ST-Filter는 인덱스를 사용함으로써 후보 서브시퀀스들을 빠르게 검색하기 때문이다.

실험 2에서는 maxWarpRatio를 다양하게 변화시키면서 Naive-Scan, LB-Scan, ST-Filter 등 기존 기법들의 성능의 변화를 관찰하였다. 사용된 max-WarpRatio는 3, 6, 9, 12이며, 질의 시퀀스의 길이는 110, 선택률은 2.13×10-7을 사용하였다.

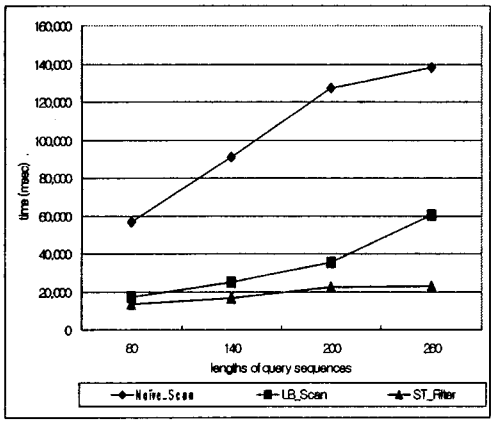
그림 4.2는 maxWarpRatio의 변화에 따르는 각 기법의 수행 시간의 변화를 그래프로 표현한 것이다. 가로축은 maxWarpRatio의 값을 나타내며, 세로축은 타임 워핑 하의 서브시퀀스 매칭을 처리할 때 소요되는 전체 시간을 나타낸다.

maxWarpRatio가 증가함에 따라 모든 경우에서 전체 수행 시간은 점차 증가하는 것으로 나타났다. maxWarpRatio가 클수록 질의 시퀀스와 비교해야 하는 경우의 수가 많아지므로, 서브시퀀스 매칭에서의 CPU 처리 시간이 증가되기 때문이다. 실험 1의 결과와 마찬가지로, ST-Filter가 모든 maxWarpRatio 값에서 가장 좋은 성능을 나타냈고, Naive-Scan이 가장 떨어지는 성능을 보였다. 실험 1의 결과에서 논의한 바와 같이, 이러한 결과는 여과 단계의 효과와 인덱스의 사용 효과에서 기인한 것이다.

실험 3에서는 다양한 길이의 질의 시퀀스에 대하여 Naive-Scan, LB-Scan, ST-Filter 등 기존 기법들을 이용한 서브시퀀스 매칭을 수행하였다. 사용된 질의 시퀀스의 길이는 80, 140, 200, 260이다. maxWarpRatio는 5로, 선택률은 2.13×10-7로 설정하였다.

그림 4.3은 질의 시퀀스 길이의 변화에 따르는 각 기법의 전체 처리 시간의 변화를 그래프로 표현한 것이다. 가로축은 질의 시퀀스의 길이를 나타내며, 세로축은 타임 워핑 하의 서브시퀀스 매칭을 처리할 때 소요되는 전체 시간을 나타낸다. 질의 시퀀스의 길이가 길어짐에 따라 모든 경우에서 전체 처리 시간은 증가하는 것으로 나타났다. 이는 서브시퀀스 s와 질의 시퀀스 q에 대한 거리 축적 테이블을 구성하는 시간이 O(|s|×|q|)로 나타나기 때문이다. 전체적인 경향은 이전의 다른 실험들에서와 마찬가지로 나타났다. 여과 단계를 사용하지 않는 Naive-Scan이 가장 떨어지는 성능을 나타냈고, 여과

단계에서 인덱스를 사용하는 Suffix-Filter가 가장 좋은 성능을 보였다. 특히, Suffix-Filter는 질의 시퀀스 길이의 변화에 큰 영향을 받지 않음을 볼 수 있었다.



▶▶ 그림 4.3 질의 시퀀스 길이의 변화에 따르는 전체 처리 시간

V. 결론

타임 워핑 하의 시퀀스 매칭은 주어진 질의 시퀀스와 타임 워핑 거리가 허용치 이하인 시퀀스들을 시계열 데이터베이스로부터 찾아내는 연산이다.

본 논문에서는 먼저 타임 워핑 하의 시퀀스 매칭을 지원하는 기존의 기법 Naive-Scan, LB-Scan, ST-Filter의 특성을 지적하고, 이들을 전체 매칭 및 서브시퀀스 매칭에 각각 적용하는 방안에 관하여 논의하였다. 또한, 실제 주식 데이터를 이용한 다양한 실험을 통하여 이들에 대한 정량적인 성능 평가를 수행하였다. 타임 워핑 하의 서브시퀀스 매칭을 위한 기존 기법들 간의 정량적인 성능을 직접 분석한 연구 결과는 아직 제시된 바 없다. 따라서 본 연구 결과는 세 가지 기법들에 대한 성능을 비교할 수 있는 중요한 자료로서 사용될 수 있을 것이다.

현재, 본 저자들은 이러한 성능 평가 결과를 기반으로 타임 워핑 하의 서브시퀀스 매칭의 성능 병목을 파악하고, 이를 최적화 할 수 있는 기법을 후속 연구 주제로 추진 중에 있다.

감사의 글

본 연구는 정보통신부 대학 IT연구센터(센터명: 미디어서비스기술연구센터) 육성·지원사업의 연구결과로 수행되었습니다.

▌참고문헌▐

[Agr93] 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.

[Agr95] 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.

[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.

[Cha84] C. Chatfield, The Analysis of Time-Series: An Introduction, 3rd Edition, Chapman and Hall, 1984.

[Cha99] 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.

[Che96] Chen, M. S., Han, J., and Yu, P. S., "Data Mining: An Overview from Database Perspective," IEEE Trans. on Knowledge and Data Engineering, Vol. 8, No. 6, pp.866-883, 1996.

[Chu99] 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.

[Das97] 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.

[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.

[Gol95] 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.

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

[Loh00] 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.

[Loh01] 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.

[Moo01] Y. S. Moon, K. Y. Whang, and W. K. Loh, "Duality-Based Subsequence Matching in Time-Series Databases," In Proc. Int'l Conf. on Data Engineering, IEEE ICDE, pp.263-272, 2001.

[Moo02] Y. S. Moon, K. Y. Whang, and W. S. Han, "GeneralMatch: A Subsequence Matching Method in Time-Series Databases Based on Generalized Windows, In Proc. Int'l. Conf. on Management of Data, ACM SIGMOD, 2002.

[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 ICDE, pp.23-32, 2000.

[Par01] 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.

[Rab93] L. Rabiner and H. H. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.

[Raf97] 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.

[Raf99] D. Rafiei, "On Similarity-Based Queries for Time Series Data," In Proc. Int'l. Conf. on Data Engineering, IEEE ICDE, pp.410-417, 1999.

[Ste94] G. A. Stephen, String Searching Algorithms, World Scientific Publishing, 1994.

[Yi98] 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.

[Yi00] B. K. Yi and C. Faloutsos, "Fast Time Sequence Indexing for Arbitrary Lp Norms," In Proc. Int'l. Conf. on Very Large Data Bases, VLDB, pp.385-394, 2000.