DTW 거리를 지원하는 범위 서브시퀀스 매칭
(Range Subsequence Matching under Dynamic Time Warping)

한 욱 신 † 이 진 수 †† 문 양 세 †††
(Wook-Shin Han) (Jinsoo Lee) (Yang-Sae Moon)

요 약 본 논문에서는 동적 타임 워핑(DTW) 거리를 사용하는 범위 서브시퀀스 질의 처리 방법을 제안한다. 본 논문에서는 제안하는 방법은 데이타 시퀀스를 디스조인트 윈도우로 분할하고, 질의 시퀀스를 슬라이딩 윈도우로 분할하는 방법을 사용하는 DualMatch의 범위 서브시퀀스 질의 처리 방법을 이용한다. DualMatch는 유클리디언 거리 하에서 동작하는 것으로 알려져 있다. 그러나, 유클리디언 거리는 견고하지 못한 유사 모델이기 때문에 DualMatch는 반드시 DTW 거리를 지원해야 한다. 본 논문에서는 제안하는 방법의 정확성을 입증하기 위해서 중요한 정리를 유도하고, 이에 근거한 알고리즘을 제안한다. 광범위한 실험을 통해 본 논문에서 제안하는 방법이 순차 스캔 알고리즘 보다 효율적으로 동작함을 보였다.

키워드 : 서브 시퀀스 매칭, 동적 타임 워핑(DTW)

Abstract In this paper, we propose a range subsequence matching under dynamic time warping (DTW) distance. We exploit Dual Match, which divides data sequences into disjoint windows and the query sequence into sliding windows. However, Dual Match is known to work under Euclidean distance. We argue that Euclidean distance is a fragile distance, and thus, DTW should be supported by Dual Match. For this purpose, we derive a new important theorem showing the correctness of our approach and provide a detailed algorithm using the theorem. Extensive experimental results show that our range subsequence matching performs much better than the sequential scan algorithm.

Key words : Subsequence matching, DTW

1. 서 론

시계열 데이타는 멀티미디어 검색, 데이타 마이닝이나 데이타 웨어하우징과 같은 데이타베이스 응용분야에서 그 중요성이 크게 증가하고 있다[1,2]. 시계열 데이타는 각 시간대 별로 측정한 실수 값의 연속이다. 시계열 데이타의 예로는 음악 데이타, 주식 데이타, 네트워크 트레픽 데이타 등이 있다. 데이타베이스에 저장된 시계열 데이타를 데이타 시퀀스라 하고, 사용자에게 유사 검색을 위해 주어진 시계열 데이타를 질의 시퀀스라 한다.

최근에 주어진 질의 시퀀스와 유사한 데이타 시퀀스를 데이타베이스에서 찾는 문제인 유사 시퀀스 매칭에 사용되는 여러 가지 유사 모델이 연구되어 오고 있다. 본 논문에서는 동적 타임 워핑(DTW)를 유사 모델로 사용한다[3,4]. 동적 타임 워핑 거리는 가장 견고한 유사 모델 중 하나로 알려져 있으며, 허밍을 이용한 질의[5], 음성 인식[6], 이미지 검색[7] 등의 많은 응용에서 사용되고 있다.

범위 서브시퀀스 매칭은 다양한 길이를 가지는 N개의 데이타 시퀀스 S_1, S_2, …, S_N과 질의 시퀀스 Q, 그리고 허용 오차 ε이 주어 졌을 때, 질의 시퀀스 Q와 하나 이상의 ε-매치인 서브시퀀스를 가지는 모든 시퀀스 S_i와 그 서브시퀀스의 오프셋을 찾는 문제이다[8]. 주어진 두 시퀀스의 거리가 ε 이하이면 두 시퀀스를 ε-매치라 한다.

본 논문에서는 우리의 이전 연구인 유클리디언 거리 하에서 효율적이며 간단한 방법으로 범위 서브시퀀스 매칭을 할 수 있는 DualMatch[8]의 윈도우 구성 방법

· 이 논문은 2005년도 한국학술진흥재단의 지원에 의하여 연구되었음 (KRF-2005-C00173)

† 종신회원 : 경북대학교 컴퓨터공학과 교수
wshan@knu.ac.kr
†† 학생회원 : 경북대학교 컴퓨터공학과
jslee@www-db.knu.ac.kr
††† 종신회원 : 강원대학교 컴퓨터과학과 교수
ysmoon@cs.kangwon.ac.kr

논문접수 : 2008년 4월 1일
심사완료 : 2008년 6월 4일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제6호(2008.8)

을 이용한다. DualMatch의 윈도우 구성방법은 데이타 시퀀스를 디스조인트 윈도우로 나누고 질의 시퀀스를 슬라이딩 윈도우로 나눈다. 본 논문에서는 DualMatch를 DTW 거리를 지원하도록 확장한다. 본 논문에서는 실제 주식 데이타와 랜덤 워크 데이타를 사용해 다양한 실험을 수행하였다. 실험 결과, 본 논문에서 제안하는 방법은 순차 스캔 방법에 비해 수 십배 효율적으로 동작하는 것을 보였다. 특히, 허용 오차 ε이 매우 작을 때 더욱 효과적으로 동작한다.

[9]에서 제안한 방법의 기본 아이디어만 제시하고 있으며, 본 논문에서는 구체적인 알고리즘을 제시하고 보다 다양한 데이타를 이용하여 실험을 수행하여 그 결과를 설명한다.

본 논문은 나머지 부분은 다음과 같이 구성된다. 제2절에서 DTW에 대해서 간략하게 설명하고 제3절에서는 관련 연구에 대해서 설명한다. 제4절에서는 DTW 거리 하에서의 범위 서브시퀀스 매칭에 대해서 설명한다. 제5절에서는 실험결과를 설명하고 이를 분석한 후, 제6절에서 결론을 맺는다.

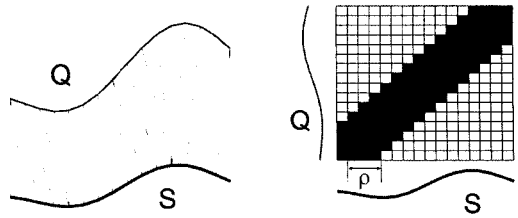
2. 동적 타임 워핑(DTW)

본 절에서는 동적 범위 서브시퀀스 매칭을 설명하기 전에 본 논문에서 사용한 유사 모델인 동적 타임 워핑 거리와 시퀀스 단계와 인덱스 단계에서 사용할 수 있는 하한 거리에 대해서 설명한다.

같은 길이를 가지는 두 시퀀스 Q와 S에 대해서 DTW 거리는 다음과 같이 재귀적으로 정의된다.

$$DTW = (<>,<>) = 0$$
$$DTW(S,<>) = DTW(<>,Q) = \infty$$
$$DTW(S,Q) = \sqrt[p]{|S[1]-Q[1]|^p + \min\begin{cases} DTW(Rest(S),Rest(Q)) \\ DTW(Rest(S),Q) \\ DTW(S,Rest(Q)) \end{cases}}$$

그림 1(a)는 DTW를 통해서 두 시퀀스 Q, S가 어떻게 정렬되는지를 보여준다. DTW 거리는 그림 1(b)와 같은 행렬을 이용하여 동적 프로그래밍(dynamic programming)을 통해 계산이 된다. 워핑 패스는 행렬의 값들의 연속으로 정의 된다. 워핑 패스 중 (i, j) 번째 행렬 값은 질의 시퀀스 Q의 i번째 값 Q[i]와 데이타 시퀀스 S의 j번째 값 S[j]가 서로 정렬되었음을 의미한다. 쿼리 시퀀스의 하나의 포인트가 과도한 수의 데이타 시퀀스의 포인트와 정렬되는 것을 방지하고 계산 시간을 줄이기 위해서 Sakoe-Chiba band[4]와 Itakura Parallelogram[10]와 같은 전역 제약사항을 사용한다. Sakoe-Chiba band에서는 ρ를 워핑 넓이라 할 때, |i-j| > ρ일 때, 행렬 값을 무한대로 설정 한다. DTWρ는 워핑 넓이가 ρ일 때의 DTW거리 값을 의한다. 여기서, 만약 ρ가



(a) DTW 비교 (b) DTW 계산을 위한 행렬

그림 1 DTW의 예

0일 경우에는 DTWρ는 유클리디언 거리와 동일하게 된다. 본 논문에서는 표현의 편의를 위해서 특별한 언급이 없는 한 DTWρ를 DTW로 표현한다.

전체 매칭을 위한 하한 거리에 대해 설명하기 앞서, 질의 봉투(query envelope)[5]와 구분적 집합 근사(piecewise aggregate approximation, PAA)[11,12]에 대해서 설명한다.

정의 1. 워핑 넓이 ρ가 주어졌을 때, 질의 Q에 대한 질의 봉투 E(Q)는 상위와 하위 봉투로 구성되며, Q의 질의 봉투는 상위 봉투 U와 하위 봉투 L사이의 영역을 나타낸다. E(Q)의 i번째 원소(L[i], U[i])는 다음과 같이 정의 된다.

$$L[i] = \min_{-\rho \le r \le \rho}(Q[i+r]), U[i] = \max_{-\rho \le r \le \rho}(Q[i+r])$$

길이가 N인 데이타 시퀀스 S의 PAA (P(S) 라고 표현함)는 길이가 f인 타임 시리즈 {S[1], …, S[f]} 로 나타낼 수 있다. 이때, S[i]는 데이타 시퀀스 S의 i번째 원소라 할 때, S[i]는 다음과 같이 정의 된다.

$$\overline{S[i]} = \frac{f}{N}\sum_{j=\frac{N}{f}(i-1)+1}^{\frac{N}{f}i} S[j]$$

이와 마찬가지로, P(E(Q))의 i(1≤i≤f)번째 원소 (L[i], U[i])는 다음과 같이 정의 된다.

$$\overline{L[i]} = \frac{f}{N}\sum_{j=\frac{N}{f}(i-1)+1}^{\frac{N}{f}i} L[j], \overline{U[i]} = \frac{f}{N}\sum_{j=\frac{N}{f}(i-1)+1}^{\frac{N}{f}i} U[j].$$

이제 질의 봉투 E(Q)와 데이타 시퀀스와의 거리(LB_Keogh라 불림)를 정의한다. LB_Keogh는 시퀀스 단계에서의 가장 타이트한 DTW의 하한 거리 중 하나이며, 다음과 같이 정의 된다.

$$LB\_Keogh(E(Q),S) = \sqrt[p]{\sum_{i=1}^{N}\begin{cases} |S[i]-U[i]|^p & if\ S[i] > U[i] \\ |S[i]-L[i]|^p & if\ S[i] < L[i] \\ 0 & otherwise \end{cases}}$$

색인 레벨에서 사용할 수 있는 하한 거리로는 LB_PAA가 있다. LB_PAA는 질의 봉투의 PAA와 데이타 시퀀스의 PAA(P(S))사이의 거리이며, 다음과 같이 정

의되며, 그림 2와 같이 도식화 할 수 있다.

$$LB\_PAA\left(P(E(Q)),P(S)\right) = \sqrt[p]{\sum_{i=1}^{f}\frac{N}{f}\begin{cases}|\overline{S[i]}-\overline{U[i]}|^p & if\ \overline{S[i]}>\overline{U[i]}\\ |\overline{S[i]}-\overline{L[i]}|^p & if\ \overline{S[i]}<\overline{L[i]}\\ 0 & otherwise\end{cases}}$$

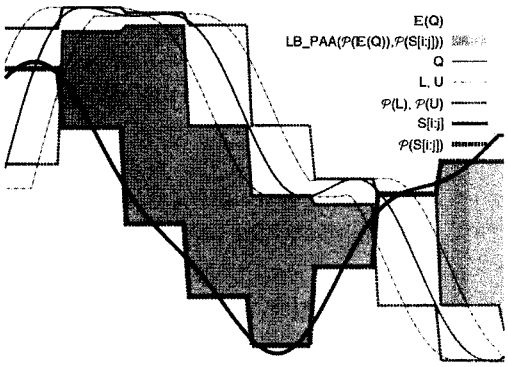


그림 2 LB_PAA의 도식화

DTW하에서 착오기각(false dismissal)이 없음을 보장하기 위해서 다음 보조 정리[5]를 정의한다.

보조정리 1. 같은 길이를 가지는 두 시퀀스 Q와 S, 워핑 넓이 ρ가 주어 졌을 때, 다음 식은 항상 참이다.

$$DTW(Q,S) \geq LB\_Keogh(E(Q),S) \geq LB\_PAA\left(P(E(Q)),P(S)\right)$$

3. 관련 연구

먼저, 대표적 범위 시퀀스 매칭인 Agrawal 등이 제안한 전체 매칭 방법[13]을 설명한다. 이 검색 방법은 크게 색인 구축 단계와 유사성 검색 단계로 나뉘어 진다. 색인 구축 단계에서는 길이가 N인 데이타 시퀀스를 저차원 변환을 하여 f차원의 점(f << N)으로 변환한다. 이때, 이 변환과정을 저차원 변환이라 한다. 그리고 저차원 변환된 각 점을 R*-트리[14]에 저장을 한다. R*-트리는 고차원의 데이타를 효율적으로 저장하고 관리하지 못하는 문제[15,16]로 인하여, 작은 수의 특성만을 추출하여 저차원의 데이타를 사용해야 한다. 이 때, 저차원 변환을 위해 사용되는 함수를 특성 추출 함수라 한다[8,11,17]. 특성 추출 함수는 다음 식을 항상 만족해야 한다[18].

$$D_1\left(F(S_1),F(S_2)\right) \leq D_2(S_1,S_2)$$

여기서, D2는 시퀀스 S1과 S2사이의 거리를 계산하는 함수이고, D1은 S1과 S2들을 특성 추출 함수 F를 사용하여 저차원 변환된 점들 사이의 거리를 계산하는 함수이다. 시퀀스의 유사 검색에 있어서, 질의 시퀀스는 위와 유사하게 f차원의 저차원 점으로 변환되며, 저차원 변환된 점과 허용 오차 ε을 사용하여 범위 질의를 구성한다. 그런 다음, 범위 질의를 사용하여 색인에서 잠정적으로 ε-매치할 가능성이 높은 후보들을 찾는다. 이 과정은 착오기각(후보에는 반드시 ε-매치인 모든 시퀀스가 포함됨)이 없음을 보장하지만, N차원이 아닌 저차원 변환된 f차원의 점을 사용하였기 때문에 착오해답(후보이나 실제로는 질의 시퀀스와 ε-매치하지 않는 서브시퀀스)을 생성할 수 있다. 따라서, 모든 후보들에 해당하는 시퀀스를 디스크로부터 읽어와 질의 시퀀스와의 거리를 계산하여 후보로부터 착오해답을 제외해야한다.

DualMatch[8,9]와 GeneralMatch[17]는 FRM과는 다른 윈도우 구성 방법을 사용하여 범위 서브시퀀스 매칭 알고리즘의 성능을 개선하였다. DualMatch는 윈도우 구성방법의 *이원성*이라는 개념을 정의하여 데이타 시퀀스를 디스조인트 윈도우로, 질의 시퀀스를 슬라이딩 윈도우로 나누고 범위 서브시퀀스 매칭을 수행한다. GeneralMatch는 디스조인트 윈도우와 슬라이딩 윈도우의 개념을 일반화 하여, J-디스조인트 윈도우와 J-슬라이딩 윈도우의 개념을 정의하였다. 윈도우 구성 방법과 색인 구성 방법을 제외하고는 DualMatch와 General-Match는 FRM의 그것과 유사하다.

그러나, 위 연구에서 제안하는 알고리즘들은 오직 유클리디언 거리 하에서만 동작을 하며, DTW거리 하에서는 동작을 하지 않는다. [19]에서는 DTW하에서의 범위 서브시퀀스 매칭 알고리즘을 소개 하였으나 자세한 구현과 질의 방법에 대한 설명이 없다. 따라서 본 논문에서는 이를 설명한다.

4. DTW 거리를 지원하는 범위 서브시퀀스 매칭

자세한 설명을 하기에 앞서 먼저, 몇 가지 용어를 정의 한다. 주어진 시퀀스 S에 대해서 만약, $i_1 \geq i_2$이고 $j_1 \leq j_2$이라면, 서브시퀀스 $S[i_1: j_1]$은 서브시퀀스 $S[i_2: j_2]$를 *포함(include)*한다 라고 정의한다. S가 유한개의 디스조인트 윈도우로 나뉘어지고, 서브시퀀스 S[i:j]에 대해서 S의 모든 디스조인트 윈도우 중 몇몇의 디스조인트 윈도우들이 포함 될 때, 이 디스조인트 윈도우들을 *포함 윈도우*(included windows)라 한다. 길이가 같은 서브시퀀스들의 포함 윈도우의 수는 시퀀스 S에서 서브시퀀스의 시작 위치에 따라 달라진다. 따라서, 알고리즘의 간단화를 위해 서브시퀀스의 위치에 관계없이 포함 윈도우의 수를 정의할 필요가 있다. *최소 포함 윈도우 개수*는 길이가 l인 서브시퀀스의 시작 위치와 관계없이, 포함 윈도우의 수 중 최소 값을 의미한다. 만약 시퀀스 S가 크기가 ω인 디스조인트 윈도우로 나누어진다면 길이가 l인 서브시퀀스의 최소 포함 윈도우 개수 r은 다음과 같이 계산된다.

$$r = \lfloor (l + 1/\omega) \rfloor - 1.$$

정리 1. 데이타 시퀀스 S는 길이가 ω인 디스조인트 윈도우에 의해서 나뉘어 지고, 질의 시퀀스 Q는 길이가 ω인 슬라이딩 윈도우로 나뉘어진다고 가정하자. 만약 길이가 Len(Q)인 S의 서브시퀀스 S[i:j]가 Q와 ε-매치라면, 적어도 하나의 S[i]으로부터 시작하는 오프셋을 가지는 S[i:j]의 포함 윈도우 s_m은 Q[1]으로부터 같은 오프셋을 가지는 Q의 슬라이딩 윈도우와 $\varepsilon/p\sqrt{r}$-매치 이다. 여기서, r은 길이가 Len(Q)인 서브시퀀스의 최소 포함 윈도우 개수이다.

증명. 보조 정리 1에 의해서 다음의 수식을 유도 할 수 있다.

$$DTW(Q, S[i:j]) \le \varepsilon \Rightarrow LB\_PAA(P(E(Q)), P(S)) \le \varepsilon$$

DualMatch의 윈도우 구성 방법에 의해 데이타 시퀀스 S[i:j]는 적어도 r개의 디스조인트 윈도우 s_1, …, s_r와 디스조인트 윈도우를 제외한 데이타 시퀀스 S[i:j]의 앞부분과 뒷부분 s_h, s_t으로 나뉘어질 수 있다. 따라서, 데이타 시퀀스 S[i:j]는 $s_h s_1 \cdots s_r s_t$로 표현될 수 있다. 이와 마찬가지로 질의 시퀀스 Q도 $q_h q_1 \cdots q_r q_t$로 표현될 수 있다. 따라서, 위의 수식은 아래의 수식으로 표현 가능하다.

$$DTW(Q, S[i:j]) \le \varepsilon \Rightarrow$$

$$LB\_PAA(P(E(q_h q_1 \ldots q_r q_t)), P(s_h s_1 \ldots s_r s_t)) \le \varepsilon$$

제2장에서 LB_PAA의 정의와 같이 LB_PAA는 단조증가 함수임으로, 위 수식의 데이타 시퀀스와 질의 시퀀스의 {s_h, s_t}, {q_h, q_t}을 제외한 다음 수식으로 유도 가능하다.

$$DTW(Q, S[i:j]) \le \varepsilon \Rightarrow LB\_PAA(P(E(q_1 \ldots q_r)), P(s_1 \ldots s_r)) \le \varepsilon$$

질의 시퀀스와 데이타 시퀀스의 매칭 윈도우 패어들 중 {(q_1, s_1), …, (q_r, s_r)} 중에서 LB_PAA(P(E(q_i)), P(s_i))의 값이 가장 작은 매칭 윈도우 패어를 (q_m, s_m)이라 한다면, LB_PAA의 정의에 의해 다음의 수식을 유도 할 수 있다.

$$LB\_PAA(P(E(q_1 \ldots q_r)), P(s_1 \ldots s_r)) \le \varepsilon$$
$$\Rightarrow \sqrt[p]{\sum_{k=1}^{r} LB\_PAA(P(E(q_m)), P(s_m))^p} \le \varepsilon$$
$$\Rightarrow LB\_PAA(P(E(q_m)), P(s_m)) \le \varepsilon/\sqrt[p]{r}$$

따라서, 두 시퀀스 S[i:j]와 Q가 ε-매치라면 S[i:j]의 디스조인트 윈도우 s_m과 $\varepsilon/\sqrt[p]{r}$-매치인 Q의 슬라이딩 윈도우 q_m 존재 한다. □

여기서, 만약 S[i:j]와 Q가 DTW 거리 하에서 ε-매치라면, 서브시퀀스 S[i:j]의 r개의 포함 윈도우 중 적어도 하나의 포함 윈도우는 LB_PAA하에서 q_m과 $\varepsilon/\sqrt[p]{r}$-매치이다.

그림 3은 본 논문에서 제안하는 범위 서브 시퀀스 매칭의 대략적인 수행 과정의 예를 보여준다. 먼저, 사용자가 제시한 질의 시퀀스 Q에 대해서 QMBR을 구성한다. 이 과정은 Q의 모든 슬라이딩 윈도우를 저차원 변환 함수(PAA)를 사용하여 저차원으로 변환 한다. 저차원 변환된 모든 점을 포함하는 최소 포함 상자(MBR)을 만든 뒤, 정리 1에 따라 이 MBR에 $\varepsilon/\sqrt[p]{r}$ 만큼 범위를 늘려 범위 질의 QMBR를 구성한다. 다음 단계로, QMBR을 사용하여 색인에 범위 스캔을 수행하여 후보들을 찾는다. 이 과정에서 찾은 후보들은 실제 해답뿐만 아니라, 착오 해답도 포함되어 있다. 따라서, 착오 해답을 제거하기 위해서 정제 과정을 거쳐 실제 ε-매치인 서브시퀀스들을 찾는다. 이 단계에서는 데이타 베이스에 접근하여 각 후보에 해당하는 서브시퀀스를 읽어 와 해당 서브시퀀스가 ε-매치인지를 검사를 한다. DTW 함수의 호출 횟수를 줄이기 위해서 시퀀스 단계에서 사용

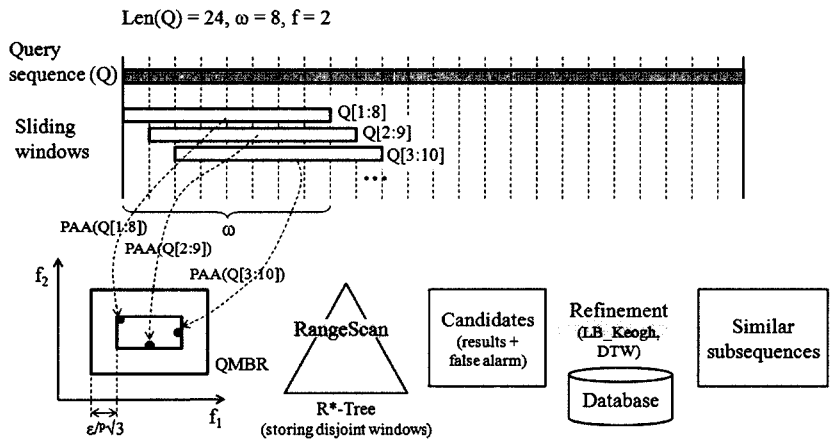


그림 3 알고리즘 RangeSearch의 예

할 수 있는 하한 거리인 LB_Keogh를 수행한 뒤, LB_Keoght 거리 하에서 ε-매치라면 DTW 거리를 계산한다.

알고리즘 1은 본 논문에서 제안하는 범위 서브시퀀스 매칭 알고리즘 RangeSearch의 자세한 구현을 나타낸다. RangeSearch 는 사용자로부터 입력 받은 질의 시퀀스 Q와 허용 오차 ε, 저차원 변환된 데이타 시퀀스의 디스조인트 윈도우들을 저장하고 있는 색인(index), 그리고 모든 데이타 시퀀스를 저장하고 있는 데이타베이스(db) 입력으로 받는다. 알고리즘은 입력 받은 Q와 ε-매치하는 서브시퀀스를 포함하는 데이타 시퀀스들의 식별자와 해당 서브시퀀스의 시작 위치를 원소로 하는 집합을 반환한다. 알고리즘 RangeSearch는 총 3개의 단계를 통해 해답을 찾는다. 첫 번째 단계는 초기화 단계로서, Q를 이용하여 질의 최소 포함 상자(QMBR)을 구성한다(행 1-8). 두 번째 단계는 색인 검색 단계로서, 첫 번째 단계에서 구성한 질의 최소 포함 상자를 사용하여 색인으로부터 후보 집합을 찾는다(행 9). 세 번째 단계는 후처리 단계로서, 두 번째 단계에서 찾은 후보 집합 중 착오 해답(false alarm)을 제거한다(행 10-16). 마지막으로 위의 세 단계를 모두 통해 찾은 해답을 반환 한다(행 17).

알고리즘 1. RangeSearch

입력: Q, ε, index, db
출력: Q와 ε-매치하는 서브시퀀스를 포함하는 데이터 시퀀스들의 식별자와 해당 서브시퀀스의 시작 위치

```
/* 1. 질의 최소 포함 상자(MBR) 구성 */
1:  for i ← 1 to Len(Q) -ω+1 do
2:    q_i ← PAA(Q[i:i+ω-1]);
3:    for j ← 1 to f do /* f: 저차원 포인트의 차원 수 */
4:      QMBR[j].l ← min(QMBR[j].l, q_i[j]);
5:      QMBR[j].h ← max(QMBR[j].h, q_i[j]);
6:  for j ← 1 to f do
7:    QMBR[j].l ← QMBR[j].l - ε/p√r;
8:    QMBR[j].h ← QMBR[j].h + ε/p√r;
/* 2. 색인 검색 단계 */
9:  candidates ← RangeScanOverIndex(index, QMBR);
/* 3. 후처리 단계: candidates 중 착오 해답을 제거 */
10: for each cand in candidates do
11:   id ← cand.id;
12:   offset ← cand.offset;
13:   sub ← ReadSubseqFromDB(db, S_id , offset, Len(Q));
14:   if LB_Keogh(P(Q), sub) ≤ ε then
15:     if DTW(Q, sub) ≤ ε then
16:       results ← results ∪ {id, offset};
17: return results;
```

5. 성능 평가

본 논문에서는 LB_Keogh를 이용한 순차 검색과 본 논문에서 설명한 범위 서브시퀀스 매칭의 성능을 비교한다. 버퍼의 크기는 데이타 베이스 크기의 5%로 설정을 한다.

DTW하에서의 범위 서브시퀀스 매칭 방법의 효율성을 입증하기 위해서 본 논문에서는 두 가지 데이타 셋을 사용하여 광범위한 실험을 수행하였다. 첫 번째 데이타 셋은 FRM과 DualMatch[8]에서 사용한 실제 주식 데이타로, 329112개의 엔트리로 구성되어 있다. 이 데이타 셋을 *STOCK_DATA*라 부른다. 두 번째 데이타 셋은 FRM과 DualMatch에서 역시 사용된 데이타로 100만개의 엔트리로 구성된 랜덤 워크 데이타이다. 이 데이타는 인공적으로 생성된 데이타로서, 시작 엔트리를 1.5로 하고, 각 엔트리에 (-0.001, 0.001) 사이의 임의의 값 하나를 더하여 다음 엔트리를 구하는 방식으로 생성된다. 이 데이타는 *WALK-DATA*라 부른다.

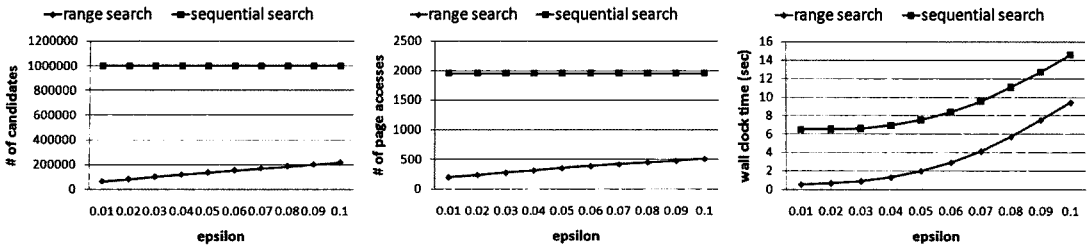
모든 실험은 512메가 바이트 메인 메모리와 Pentium IV 2.8 GHz CPU를 가지고 있는 Linux Kernel 2.6 PC에서 수행 되었다. 버퍼 페이지 재 할당 알고리즘으로는 LRU를 사용하고, 페이지 크기를 4096바이트로 설정하였다. OS 파일 시스템의 버퍼링 효과와 실제 디스크 I/O를 보장하기 위해서 O_DIRECT 플래그[20]를 사용하여 데이타 파일과 인덱스 파일을 열어서 사용하였다. 저차원 변환 함수로는 PAA를 사용하고 하나의 디스조인트 윈도우를 8차원의 데이타로 저차원 변환 시켰다. 워핑 넓이는 질의 시퀀스 길이의 5%로 설정하였다.

실험 분석을 위해 본 논문에서는 제안하는 범위 검색과 순차 스캔 알고리즘의 후보의 수, 디스크 페이지 접근 횟수, 그리고 수행시간을 측정하였다. 실험에 데이타 시퀀스의 임의의 지점에서 Len(Q)길이만큼 가져와 질의 시퀀스로 사용하였다[18]. 실험결과의 잡음을 줄이기 위해서 같은 길이를 가지는 10개의 서로 다른 질의 시퀀스를 사용하여 실험결과로는 각 질의의 측정 결과의 평균 값을 사용하였다.

실험 1(허용 오차 변화 실험): 그림 4은 WALK-DATA에 대한 허용 오차 ε변화에 따른 실험 결과를 나타낸다. 그림 4(a)는 후보의 수를 나타내며 그림 4(b)는 디스크 페이지 접근 횟수를 나타낸다. 그리고, 그림 4(c)는 수행시간을 나타낸다.

그림 4(a)에서 볼 수 있듯이, 범위 검색은 순차 검색과 비교하여 후보의 수를 15.6배 감소시켰다. 순차 검색은 데이타 베이스를 모두 검색해야 함으로 항상 같은 수의 후보 수를 보인다. ε값이 증가함에 따라, 범위 스캔의 후보 수는 증가함을 보인다.

디스크 페이지 접근 횟수 측면에서, 범위 검색은 순차 검색과 비교하여 최대 10.0배의 성능 향상을 보인다. 그림 4(b)에서 볼 수 있듯이, 범위 검색은 모든 데이타 베



(a) 후보 수 (b) 디스크 페이지 접근 수 (c) 수행 시간

그림 4 WALK-DATA의 허용 오차 ε변화에 대한 실험 결과

이스를 검색해야 함으로 항상 같은 수의 디스크 페이지 접근 횟수를 보인다.

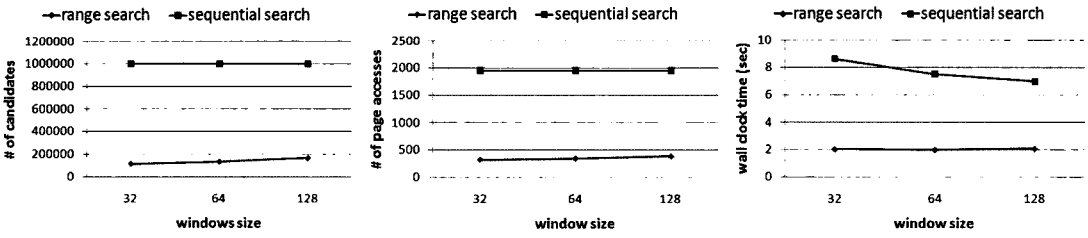
그림 4(c)에서 볼 수 있듯이, 범위 검색은 순차 검색과 비교하여 최대 12.5배 수행시간을 감소 시킨다. ε이 증가 함에 따라 DTW함수의 호출 횟수가 증가하기 때문에 순차 검색의 수행 시간이 증가하게 된다. 보다 자세하게 설명하면, 서브 시퀀스 S에 대한 DTW 함수는 서브 시퀀스 S와 질의 시퀀스와의 LB_Keogh 거리가 ε 이하일 때만 호출 된다.

실험 2(윈도우 크기 변화 실험): 그림 5은 WALK-DATA에 윈도우 크기 변화에 따른 실험 결과를 나타낸다. 그림 5(a)는 후보의 수를 나타내며 그림 5(b)는 디스크 페이지 접근 횟수를 나타낸다. 그리고, 그림 5(c)는 수행시간을 나타낸다.

범위 검색은 윈도우 크기가 커질수록 저차원 변환 함수에 의해 잃는 정보의 양이 많아 져 색인 단계에서의 후보 수가 증가 하게 된다. 이에 반해, 순차 스캔은 모든 데이타 베이스를 검색함으로 같은 수의 후보 수를 보인다.

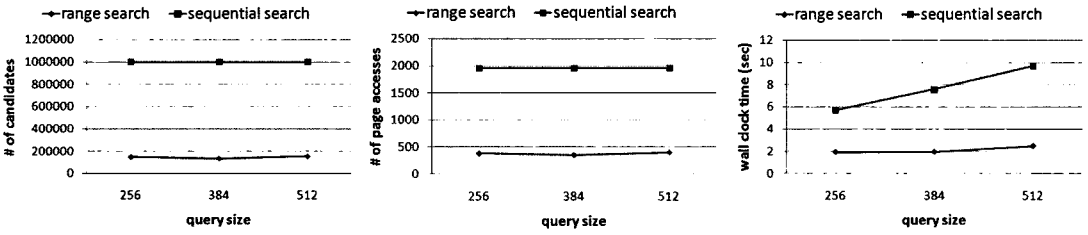
실험 3(질의 크기 변화 실험): 그림 6은 WALK-DATA에 대한 질의 크기 변화에 따른 실험 결과를 나타낸다. 그림 6(a)는 후보의 수를 나타내며 그림 6(b)는 디스크 페이지 접근 횟수를 나타낸다. 그리고, 그림 6(c)는 수행시간을 나타낸다.

그림 6(a)와 6(b)에서 볼 수 있듯이, 질의 크기가 증가함에도 범위 검색과 순차 검색은 항상 비슷한 후보 수와 디스크 페이지 접근 수를 보인다. 그러나, 그림 6(c)에서 볼 수 있듯이, 질의의 크기가 증가 할수록 수행 시간이 증가하는 것을 확인 할 수 있다. 이는, 비슷한 후보 수를 가지더라도 질의 크기가 클수록 DTW 함



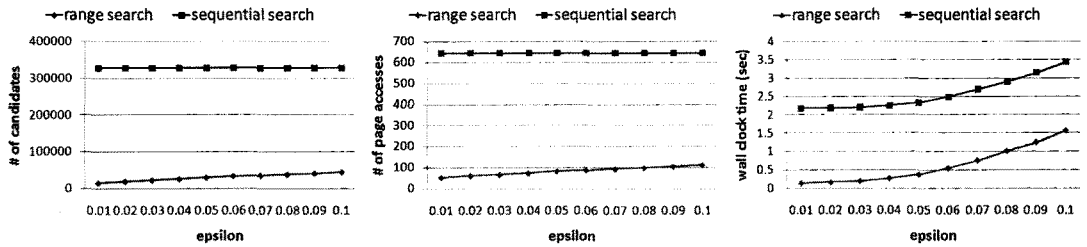
(a) 후보 수 (b) 디스크 페이지 접근 수 (c) 수행 시간

그림 5 WALK-DATA의 윈도우 크기 변화에 대한 실험 결과



(a) 후보 수 (b) 디스크 페이지 접근 수 (c) 수행 시간

그림 6 WALK-DATA의 질의 크기 변화에 대한 실험 결과



(a) 후보 수 (b) 디스크 페이지 접근 수 (c) 수행 시간

그림 7 STOCK-DATA에 대한 실험 결과

수 호출의 비용이 커지기 때문이다. 범위 검색은 순차 스캔에 비해 최대 3.89배 수행 시간을 증가 시킨다.

그림 7은 STOCK-DATA에 허용 오차 ε의 변화에 대한 실험 결과를 나타낸다. STOCK-DATA 실험 결과의 경향은 WALK-DATA의 실험 결과의 그것과 유사하다 따라서, 자세한 실험 분석은 생략한다.

6. 결 론

본 논문에서는 동적 타임 워핑(DTW) 거리를 지원하는 범위 서브시퀀스 매칭을 제안하였다. 또한, 본 논문에서는 데이타 시퀀스를 디스조인트 윈도우로 나누고 질의 시퀀스를 슬라이딩 윈도우로 나누는 DualMatch의 윈도우 구성 방법을 사용하였다. 제안한 알고리즘의 정확성을 입증하기 위해 새로운 정리를 유도하였다. 광범위 한 실험 결과 본 논문이 제안하는 범위 서브시퀀스 매칭은 순차 검색 알고리즘에 비해 우수함을 보였다.

본 연구에서 제안한 범위 서브시퀀스 알고리즘은 견고한 유사 모델인 DTW거리를 지원함으로 여러 응용에서 실제로 사용가능 할 수 있고, 실험 결과에서 볼 수 있듯이 성능 향상을 기대 할 수 있다. 향후 본 연구를 확장, 실제 응용에 적용하여 뮤직 데이타에서 허밍을 이용한 질의 처리 방법에 대한 연구를 계속할 예정이다.

참 고 문 헌

[1] Keogh, E., "A Decade of Progress in Indexing and Mining Large Time Series Databases," In VLDB, Tutorial, 2006.

[2] Rafiei, D. et al., "Querying Time Series Data Based on Similarity," IEEE TKDE, Vol.12, No.5, 2000.

[3] Berndt, D. and Clifford, J., "Finding Patterns in Time Series: a Dynamic Programming Approach," AAAI/MIT, 1996.

[4] Sakoe, H. and Chiba, S., "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," IEEE ASSP, 1978.

[5] Zhu, Y. and Shasha, D., "Warping Indexes with Envelope Transforms for Query by Humming," In SIGMOD, 2003.

[6] Rabiner, L and Juang, B., Fundamentals of Speech Recognition, Englewood Cliffs, N. J., 1993.

[7] Bartolini, I., Ciaccia, P., and Patella, M., "WARP: Accurate Retrieval of Shapes Using Phase of Fourier Descriptors and Time Warping Distance," IEEE PAMI, pp. 142-147, 2005.

[8] Moon, Y.-S., Whang, K.-Y., and Loh, W.-K., "Duality-Based Subsequence Matching in Time-Series Databases," In ICDE, pp. 263-272, 2001.

[9] Lee, J., Han W.-S., and Moon Y.-S., "Range Search under Dynamic Time Warping," In APIS, pp. 67-70, 2007.

[10] Itakura, F., "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. on ASSP, Vol. ASSP-23, No.1, pp. 67-72, 1975.

[11] Keogh, E., "Exact indexing of dynamic time warping," In VLDB, pp 406-417, 2002.

[12] Yi, B.-K. and Faloutsos, C., "Fast Time Sequence Indexing for Arbitrary Lp Norms," In VLDB, pp. 385-394, 2000.

[13] Agrawal, R., Faloutsos, C., and Swami, A., "Efficient Similarity Search in Sequence Databases," In FODO, 1993.

[14] Beckmann, N. et al., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," In SIGMOD, pp. 322-331, 1990.

[15] Berchtold, S., Bohm, C., and Kriegel, H. P., "The Pyramid-Technique: Towards Breaking the Curse of Dimensionality," In SIGMOD, pp. 142-153, 1998.

[16] Weber, R. et al., "A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces," In VLDB, pp. 194-205, pp. 194-205, Aug. 1998.

[17] Moon, Y.-S., Whang, K.-Y., and Han, W.-S., "General Match: A Subsequence Matching Method in Time-Series Databases Based on Generalized Windows," In SIGMOD, 2002.

[18] Faloutsos, C. et al., "Fast Subsequence Matching in Time-Series Databases," In SIGMOD, 1994.

[19] Han, W.-S. et al., "Ranked subsequence matching in time-series databases," In VLDB, pages 423-434,

2007.
[20] Sobell, M. O., Practical Guide to Linux Commands, Editors, and Shell Programming, Prentice Hall, 2005.

한 욱 신
1994년 경북대학교 컴퓨터공학과 졸업(공학사). 1996년 한국과학기술원 전산학과 졸업(이학석사). 2001년 한국과학기술원 전산학과 졸업(공학박사). 2003년~현재 경북대학교 컴퓨터공학과 조교수

이 진 수
2006년 경북대학교 컴퓨터공학과 졸업(공학사). 2008년 경북대학교 컴퓨터공학과 졸업(공학석사). 2008년~현재 경북대학교 컴퓨터공학과 박사과정

문 양 세
1991년 2월 한국과학기술원 과학기술대학 전산학과 학사. 1993년 2월 한국과학기술원 전산학과 석사. 2001년 8월 한국과학기술원 전자전산학과 전산학전공 박사. 1993년 2월~1997년 2월 현대전자산업(주) 통신사업본부 주임연구원. 2001년 9월~2002년 2월 (주)현대시스콤 호처리개발실 선임연구원 2002년 2월~2005년 2월 (주)인프라밸리 기술연구소 기술위원(이사). 2005년 3월~현재 한국과학기술원 첨단정보기술연구센터 연구원. 2005년 3월~현재 강원대학교 컴퓨터과학과 조교수. 관심분야는 Data Mining, Knowledge Discovery, Stream Data, Storage System, Database Applications, Mobile/Wireless Communication Services & Systems