시계열 데이터베이스에서 단일 색인을 사용한 정규화 변환 지원 서브시퀀스 매칭

문 양 세[†] · 김 진 호[††] · 노 웅 기[†††]

요 약

정규화 변환은 시계열 시퀀스를 구성하는 엔트리들의 전체적인 패턴을 분석하는데 매우 유용하다. 본 논문에서는 단일 색인을 사용한 정규화 변환 지원 서브시퀀스 매칭 방법을 제안한다. 기존의 정규화 변환 지원 서브시퀀스 매칭 방법은 다양한 길이의 질의 시퀀스를 지원하기 위하여 여러 개의 색인을 생성해야 하고, 이에 따라 색인 저장 공간의 오버헤드와 색인 관리의 오버헤드가 발생한다. 본 논문에서는 하나의 색인을 사용하면서도 다양한 길이의 질의 시퀀스에 대한 정규화 변환을 지원하는 효율적인 서브시퀀스 매칭 방법을 제안한다. 이를 위하여, 우선 정규화 변환을 일반화한 **포함-정규화 변환**(*inclusion-normalization transform*) 개념을 제시한다. 포함-정규화 변환이란 색인에 저장할 윈도우에 대해서 해당 윈도우를 포함하는 서브시퀀스의 평균과 표준편차로 정규화하는 것으로서, 기본적인 정규화 변환을 윈도우 및 서브시퀀스 개념을 사용하여 확장한 것이다. 다음으로, 포함 정규화 변환을 기존 서브시퀀스 매칭 연구에 적용하기 위한 이론적 근거를 정리로서 제시하고 증명한다. 그리고, 이 방안을 구현하기 위한 색인 구성 알고리즘 및 서브시퀀스 매칭 알고리즘을 각각 제시한다. 실제 주식 데이터에 대한 실험 결과, 제안한 방법은 기존 방법에 비해 최대 2.5~2.8배까지 성능을 향상 시킨 것으로 나타났다. 본 논문에서 제안한 정규화 변환 지원 서브시퀀스 매칭은 정규화 변환 이외의 다른 변환을 지원하는 서브시퀀스 매칭으로 일반화 될 수 있다. 따라서, 제안한 방법은 정규화 변환을 포함하는 많은 다른 종류의 변환을 지원하는 서브시퀀스 매칭에 폭넓게 적용될 수 있는 좋은 연구결과라 사료된다.

키워드 : 데이터 마이닝, 시계열 데이터베이스, 서브시퀀스 매칭, 정규화 변환

A Single Index Approach for Subsequence Matching that Supports Normalization Transform in Time-Series Databases

Yang-Sae Moon[†] · Jinho Kim[††] · Woong-Kee Loh[†††]

ABSTRACT

Normalization transform is very useful for finding the overall trend of the time-series data since it enables finding sequences with similar fluctuation patterns. The previous subsequence matching method with normalization transform, however, would incur index overhead both in storage space and in update maintenance since it should build multiple indexes for supporting arbitrary length of query sequences. To solve this problem, we propose a single index approach for the *normalization transformed subsequence matching* that supports arbitrary length of query sequences. For the single index approach, we first provide the notion of *inclusion-normalization transform* by generalizing the original definition of normalization transform. The inclusion-normalization transform normalizes a window by using the mean and the standard deviation of a subsequence that includes the window. Next, we formally prove correctness of the proposed method that uses the inclusion-normalization transform for the normalization transformed subsequence matching. We then propose subsequence matching and index building algorithms to implement the proposed method. Experimental results for real stock data show that our method improves performance by up to 2.5~2.8 times over the previous method. Our approach has an additional advantage of being generalized to support many sorts of other transforms as well as normalization transform. Therefore, we believe our work will be widely used in many sorts of transform-based subsequence matching methods.

Key Words : Data Mining, Time-Series Database, Subsequence Matching, Normalization Transform

1. 서 론

시계열 데이터(time-series data)란 각 시간별로 측정한 실수 값의 시퀀스로, 그 예로는 주식 데이터, 환율 데이터, 날씨 변동 데이터 등이 있다[1, 6, 16]. 시계열 데이터베이스에 저장된 시계열 데이터를 **데이터 시퀀스**(*data sequence*)라 부르며, 사용자에 의해 주어진 시퀀스를 **질의 시퀀스**(*query sequence*)라 부른다. 그리고, 주어진 질의 시퀀스와 유사한 데이터 시퀀스를 검색하는 방법을 **유사 시퀀스 매칭**(*similar sequence matching*)이라 한다[1,6]. 일반적으로,

※ 본 연구는 첨단정보기술연구센터를 통하여 과학기술부/한국과학재단의 지원을 받았음.
† 정 회 원 : 강원대학교 IT특성화대학 컴퓨터학부 조교수
†† 정 회 원 : 강원대학교 IT특성화대학 컴퓨터학부 교수
††† 정 회 원 : 한국과학기술원 전자전산학과/첨단정보기술연구센터 초빙교수
논문접수 : 2005년 12월 31일, 심사완료 : 2006년 5월 18일

유사 시퀀스 매칭에서는 길이 n인 두 시퀀스 X(=$\{X[1], X[2],\cdots, X[n]\}$)와 Y(=$\{Y[1], Y[2],\cdots, Y[n]\}$)의 거리가 사용자가 제시한 **허용치**(*tolerance*)인 ε 이하이면, 두 시퀀스 X와 Y는 **유사**(*similar*)하다고 정의한다[1,6,11]. 그리고, 본 논문에서는 거리 함수 $D(X,Y)$로 유클리디안 거리 함수 ($=\sqrt{\sum_{i=1}^{n}(X[i]-Y[i])^2}$)를 사용하며[1, 4, 6, 9], $D(X,Y)$가 ε 이하이면 X와 Y는 **ε-매치**(*ε-match*)한다고 정의한다[10, 11].

유사 시퀀스 매칭은 크게 전체 매칭(whole matching)과 서브시퀀스 매칭(subsequence matching)의 두 가지로 구분한다[6]. 전체 매칭은 질의 시퀀스와 유사한 데이터 시퀀스를 찾는 문제로서, 질의 시퀀스와 데이터 시퀀스의 길이가 동일한 특징을 갖는다[1]. 반면에, 서브시퀀스 매칭은 데이터 시퀀스에 포함된 서브시퀀스들 중에서 질의 시퀀스와 유사한 서브시퀀스를 찾는 문제로서, 사용자는 임의 길이의 시퀀스를 질의 시퀀스로 사용할 수 있다. 서브시퀀스 매칭은 전체 매칭을 일반화한 것으로, 보다 많은 응용 분야를 가진다[6, 8-11, 16]. 그리고, 유클리디안 거리 함수가 갖는 문제점을 보완하기 위하여, 정규화(normalization)[9, 15], 이동 평균(moving average)[8, 15], 쉬프팅 및 스케일링(shifting & scaling)[2, 5, 14], 타임 워핑(time warping)[7, 13, 17] 등의 많은 변환 기법이 사용되었다. 본 논문에서는 이중 정규화 변환을 지원하는 유사 시퀀스 매칭 문제를 다룬다. **정규화 변환**(*normalization transform*)은 시퀀스 엔트리들의 평균과 표준편차를 구한 후, 각 엔트리에서 평균을 빼고 표준편차로 나누는 변환이다. 이러한 정규화 변환은 절대적인 유클리디안 거리에 관계없이 시퀀스를 구성하는 엔트리들의 패턴을 분석하는데 매우 유용하다[9, 15]. 정규화 변환의 정의를 포함하여, 본 논문에서 사용하는 주요 표기와 이에 대한 정의 및 의미는 <표 1>과 같다.

〈표 1〉 정규화 변환의 정의 및 주요 표기법

기 호	정의/의미
$Len(S)$	시퀀스 S의 길이
$S[k]$	시퀀스 S의 k번째 엔트리
$S[i:j]$	시퀀스 S의 i번째에서 j번째 엔트리까지로 구성된 서브시퀀스
$\mu(S)$, $\sigma(S)$	시퀀스 S에 포함된 모든 엔트리의 평균과 표준편차
$\overline{S}$	시퀀스 S를 정규화 변환한 시퀀스 $\left(\overline{S}[i]=\frac{S[i]-\mu(S)}{\sigma(S)}\right)$
$\overline{S}[k]$	정규화 변환된 시퀀스 $\overline{S}$의 k번째 엔트리
$\overline{S}[i:j]$	정규화 변환된 시퀀스 $\overline{S}$의 i번째에서 j번째 엔트리까지로 구성된 서브시퀀스
$\overline{S[i:j]}$	서브시퀀스 $S[i:j]$를 ($\mu(S[i:j])$와 $\sigma(S[i:j])$으로) 정규화 변환한 시퀀스
s_i	시퀀스 S의 i번째 디스조인트 윈도우($=S[(i-1)\cdot\omega+1:i\cdot\omega], i\geq 1$)
$\overline{s}_i$	정규화 변환된 시퀀스 $\overline{S}$의 i번째 디스조인트 윈도우($=\overline{S}[(i-1)\cdot\omega+1:i\cdot\omega], i\geq 1$)

본 논문에서는 서브시퀀스 매칭에서 정규화 변환을 지원하는 유사 시퀀스 모델[9]을 다룬다. 즉, 질의 시퀀스 Q와 데이터 서브시퀀스 $S[i:j]$의 거리를 비교하는 것이 아니라, 이들을 정규화 변환한 이후의 두 시퀀스 $\overline{Q}$와 $\overline{S[i:j]}$의 거리를 비교하여, 변환된 두 시퀀스가 ε-매치하는지의 여부를 판단하는 유사 시퀀스 모델을 다룬다. 이와 같이 정규화 변환한 이후에 서브시퀀스 매칭을 수행하는 방법을 본 논문에서는 **정규화 변환 서브시퀀스 매칭**(*normalization transformed subsequence matching*)이라 정의한다. 다시 말해서, 정규화 변환 서브시퀀스 매칭이란 질의 시퀀스 Q와 허용치 ε이 주어졌을 때, 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$중에서 $D(\overline{Q}, \overline{S[i:j]})$가 허용치 ε 이하인 서브시퀀스 $S[i:j]$를 찾는 문제이다[9].

정규화 변환 서브시퀀스 매칭에 대해서는 Loh 등[9]이 정형적인 알고리즘이 제안하였다. Loh 등은 시퀀스를 일정한 크기의 윈도우로 나누어 정규화 변환한 후, 이들 변환한 윈도우를 사용하여 다양한 질의 시퀀스 길이에 대한 정규화 변환 서브시퀀스 매칭을 수행하였다. 그런데, Loh 등의 방법은 질의 시퀀스 길이가 정규화 변환에 사용한 윈도우 크기보다 커질 경우 성능이 크게 저하되는 문제점이 있다. 이러한 단점을 해결하기 위하여, Loh 등은 여러 크기의 윈도우에 대해서 여러 개의 다차원 색인을 생성하는 색인 보간법(index interpolation)[8]을 제안하였다. 그러나, 색인 보간법은 여러 색인을 생성함으로써 많은 저장 공간을 필요로 할 뿐 아니라, 여러 색인의 유지로 인한 관리 오버헤드가 발생하는 문제점이 있다. 또한, 질의 시퀀스 길이가 윈도우 크기보다 큰 일부 경우에 대해서는 색인을 사용하지 못하고 순차 스캔을 해야 하는 경우도 발생한다.

본 논문에서는 하나의 색인을 사용하면서도 다양한 길이의 질의 시퀀스에 대한 정규화 변환 서브시퀀스 매칭을 효율적으로 수행하는 새로운 접근법을 제안한다. 이를 위해 우선, 윈도우 $S[a:b]$에 대해서, $S[a:b]$ 자체의 평균과 표준편차가 아닌 $S[a:b]$를 포함하는 서브시퀀스 $S[i:j]$($i \leq a < b \leq j$)의 평균과 표준편차로 정규화하는 **포함-정규화 변환**(*inclusion-normalization transform*) 개념을 제시한다. 그리고, 이러한 포함-정규화 변환을 사용하여 다차원 색인을 구성하면, 하나의 색인을 사용해서도 정규화 변환 서브시퀀스 매칭을 수행할 수 있음을 보인다. 즉, 데이터 시퀀스를 나눈 각 윈도우에 대해서, 해당 윈도우를 포함하는 다양한 길이의 서브시퀀스로 포함-정규화 변환한 윈도우들을 생성하고, 이들 윈도우로 하나의 다차원 색인을 구성한다. 그런 다음, 구성한 다차원 색인을 사용하면 다양한 길이의 질의 시퀀스에 대한 정규화 변환 서브시퀀스 매칭을 지원할 수 있다.

다음으로, 포함-정규화 변환을 서브시퀀스 매칭의 기존 연구인 Faloutsos 등의 연구[2](간략히, **FRM**이라 한다)에 적용한 새로운 정규화 변환 서브시퀀스 매칭 방법을 제안한다. 이를 위하여, 우선 포함-정규화 변환의 개념을 사용하면, FRM에서 사용되었던 정리들을 그대로 활용하여 정규화 변환 서브시퀀스 매칭을 수행할 수 있음을 보인다. 특히, 데

이터 시퀀스를 나눈 각각의 윈도우에 포함-정규화 변환을 적용하는 FRM 기반의 새로운 접근법이 정규화 변환 서브시퀀스 매칭을 정확하게 수행함을 정리로서 제시하고 증명한다. 또한, FRM 기반의 정규화 변환 서브시퀀스 매칭을 구현하기 위한 색인 구성 알고리즘 및 서브시퀀스 매칭 알고리즘을 각각 제시한다. 실제 주식 데이터에 대한 실험 결과, 제안한 방법은 Loh 등의 기존 방법[9]에 비해 최대 2.5~2.8배까지 성능을 향상 시킨 것으로 나타났다.

본 논문의 구성은 다음과 같다. 제2장은 관련 연구를 설명한다. 제3장에서는 기존 연구의 문제점과 이를 해결하기 위한 본 연구의 동기를 설명한다. 제4장에서는 제안하는 단일 색인 기반의 정규화 변환 서브시퀀스 매칭을 개념, 색인 구성 알고리즘, 서브시퀀스 매칭 알고리즘의 순으로 설명한다. 제5장에서는 실험을 통해 제안한 방법의 우수성을 보이고, 마지막으로 제6장에서 결론을 맺는다.

2. 관련 연구

본 장에서는 기존의 서브시퀀스 매칭 방법을 설명한다. 서론에서 언급한 바와 같이, 정규화 변환을 제외한 다른 유사 시퀀스 모델에 대해서는 참고문헌 [2, 5, 7, 8, 13-15, 17]의 연구를 참조한다.

기존의 서브시퀀스 매칭 방법들은 Agrawal 등[1]의 전체 매칭을 발전시켜 문제를 해결하였다. 그러므로, 우선 전체 매칭을 색인 구성 알고리즘과 유사 시퀀스 매칭 알고리즘으로 구분하여 설명한다. 색인 구성 알고리즘에서는 길이 n인 데이터 시퀀스에서 f개($<< n$)의 특성(feature)을 추출하여 f-차원 공간의 점으로 **저차원 변환**(*lower-dimensional transformation*)[1, 6]한 후, 이를 f-차원의 R*-트리[3]에 저장한다. 다음으로, 유사 시퀀스 매칭 알고리즘에서는 질의 시퀀스를 데이터 시퀀스와 동일한 방법으로 f-차원 점으로 변환하고, 변환한 점과 허용치 ε을 사용하여 범위 질의(range query)를 구성한다. 그리고, 범위 질의로 R*-트리를 검색하여, ε-매치하는 모든 점들을 찾아 **후보**(*candidate*, 질의 시퀀스와 ε-매치할 가능성이 높은 데이터 시퀀스) 집합을 구한다. 이렇게 후보 집합을 구하면 **착오기각**(*false dismissal*, 유사 시퀀스이나 착오로 인해 기각되는 데이터 시퀀스)은 발생하지 않지만, 시퀀스 길이 n대신 f개의 특성만을 사용함으로 인하여 **착오해답**(*false alarm*, 후보이나 실제로는 질의 시퀀스와 ε-매치하지 않는 데이터 시퀀스)이 발생할 수 있다. 따라서, R*-트리에 대한 검색 결과로 얻은 각 후보 시퀀스들에 대해서는 데이터베이스에 저장된 실제 데이터 시퀀스를 액세스하고 질의 시퀀스와의 거리를 조사하여 착오해답을 제거하는데, 이 과정을 **후처리 과정**(*post-processing step*)이라 한다[1].

Faloutsos 등[6]은 전체 매칭을 일반화한 서브시퀀스 매칭 문제를 처음 소개하고, 이의 해결책(FRM)을 제시하였다. FRM에서는 데이터 시퀀스를 슬라이딩 윈도우로 나누고 질의 시퀀스를 디스조인트 윈도우로 나누는 방법을 사용하며, 전체 매칭과 마찬가지로 색인 구성 알고리즘과 서브시퀀스 매칭 알고리즘으로 구성된다. 먼저, 색인 구성 알고리즘에서는 데이터 시퀀스를 나눈 슬라이딩 윈도우로 f-차원의 점으로 변환하여 다차원 색인인 R*-트리에 저장한다. 그런데, 데이터 시퀀스를 슬라이딩 윈도우로 나누기 때문에 너무 많은 점이 생성되는 문제점이 있다[6,9]. 이를 해결하기 위하여, FRM에서는 여러 개의 점을 포함하는 MBR(minimum bounding rectangle)을 구성하고, 이 MBR만을 다차원 색인인 R*-트리에 저장하는 방법을 사용한다.

다음으로, FRM의 서브시퀀스 매칭 알고리즘에서는 다음 두 가지 보조정리에 기반하여 질의 시퀀스를 디스조인트 윈도우로 나누어 검색하는 방법을 사용한다.

[보조정리 1[6]] 동일한 길이의 시퀀스 S와 Q가 ε-매치한다면, 적어도 하나 이상의 디스조인트 윈도우 쌍 (s_i,q_i)가 $\varepsilon/\sqrt{p}$ -매치한다($1\le i\le p$, $p=\lfloor Len(Q)/\omega \rfloor$). 즉, 다음 식 (1)이 성립한다.

$$D(S,Q) \le \varepsilon \;\Rightarrow\; \bigvee_{i=1}^{p} D(s_i,q_i) \le \varepsilon/\sqrt{p} \qquad (1)\ \square$$

[보조정리 2[6]] 동일한 길이의 시퀀스 S와 Q가 ε-매치하면, $(S[i:j], Q[i:j])$인 어떠한 서브시퀀스 쌍도 ε-매치한다. 즉, 다음 식 (2)가 성립한다.

$$D(S,Q) \le \varepsilon \;\Rightarrow\; D(S[i:j],Q[i:j]) \le \varepsilon \qquad (2)\ \square$$

보조정리 1과 2에 따라, FRM은 질의 시퀀스를 디스조인트 윈도우로 나누어 f-차원의 점으로 변환하고, 이 점과 $\varepsilon/\sqrt{p}$ 으로 범위 질의를 구성한다. 그런 다음, R*-트리를 검색하여 $\varepsilon/\sqrt{p}$ -매치하는 MBR들을 찾아내고, 이들 MBR이 나타내는 서브시퀀스들로 후보 집합을 구성한다. 마지막으로, 후처리 과정을 통하여 착오해답을 제거하고 유사 서브시퀀스만을 찾는다.

DualMatch[10]와 GeneralMatch[11]는 윈도우 구성법을 달리하여 FRM의 성능을 개선한 서브시퀀스 매칭 방법들이다. DualMatch는 윈도우 구성에 있어서 FRM의 이원적 접근법에 해당하고, GeneralMatch는 윈도우 구성에 있어서 FRM과 DualMatch를 일반화한 접근법이다. 이들 서브시퀀스 매칭 방법의 색인 구성 및 서브시퀀스 매칭 알고리즘은 윈도우 구성을 달리하는 것을 제외하고는 FRM의 알고리즘과 유사하다.

Loh 등은 FRM을 포함한 기존 서브시퀀스 매칭 방법이 정규화 변환 서브시퀀스 매칭에 그대로 적용될 수 없음을 보였다[9]. 그 이유는 FRM에서 성립하던 보조정리 1과 2가 다음 식 (3) 및 (4)와 같이 정규화 변환 서브시퀀스 매칭에서는 성립하지 않기 때문이다.

$$D(\bar{S},\bar{Q}) \le \varepsilon \;\not\Rightarrow\; \bigvee_{i=1}^{p} D(\overline{S[(i-1)\cdot\omega+1:i\cdot\omega]},\overline{Q[(i-1)\cdot\omega+1:i\cdot\omega]}) \le \varepsilon/\sqrt{p} \qquad (3)$$

$$D(\bar{S},\bar{Q}) \le \varepsilon \;\Rightarrow\; D(\overline{S[i:j]},\overline{Q[i:j]}) \le \varepsilon \qquad (4)$$

식 (3)의 의미는 두 시퀀스 S와 Q를 정규화 변환한 시퀀스 $\bar{S}$와 $\bar{Q}$가 ε-매치한다 할지라도, S와 Q의 디스조인트 윈도우 쌍 (s_i,q_i)를 정규화 변환한 ($\overline{S[(i-1)\cdot\omega+1:i\cdot\omega]}$[1], $\overline{Q[(i-1)\cdot\omega+1:i\cdot\omega]}$)들은 어느 하나도 $\varepsilon/\sqrt{p}$-매치 하지 않을 수 있다는 것이다. 또한, 식 (4)의 의미는 정규화 변환된 두 시퀀스 $\bar{S}$와 $\bar{Q}$가 ε-매치한다 할지라도, 정규화 변환된 두 서브시퀀스 $\overline{S[i:j]}$[2]와 $\overline{Q[i:j]}$가 ε-매치하지 않을 수 있다는 것이다. 결국, FRM을 그대로 확장한 방법(정규화 변환한 데이터 윈도우들을 저차원 변환하여 다차원 색인에 저장하고, 마찬가지로 정규화 변환한 질의 윈도우를 저차원 변환하여 다차원 색인을 검색하는 방법)은 보조정리 1 및 2가 성립하지 않기 때문에 정규화 변환 서브시퀀스 매칭을 바르게 수행하지 못한다[9].

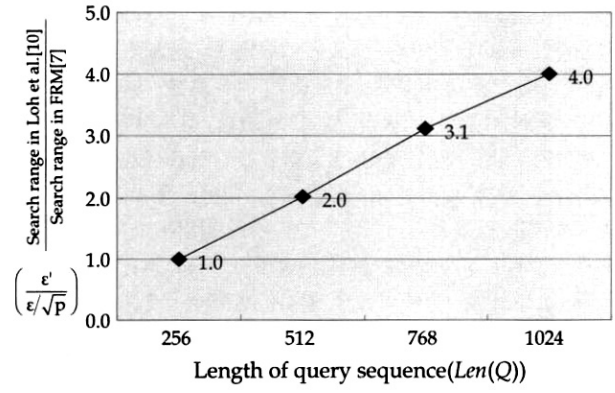
Loh 등[9]은 이러한 문제점을 해결하는 새로운 정규화 변환 서브시퀀스 매칭 방법을 제안하였다. 이들은 다음 식 (5)가 성립함을 증명하고, 보조정리 1과 2 대신 다음 식 (5)를 사용하였다.

$$D(\bar{S},\bar{Q}) \le \varepsilon \Rightarrow D(\overline{S[i:j]},\overline{Q[i:j]}) \le \varepsilon' \qquad (5)$$

식 (5)에서 ε'은 $\sqrt{2\omega - 2\sqrt{\omega^2-\omega\cdot\varepsilon^2\cdot\frac{\sigma^2(Q)}{\sigma^2(Q[i:j])}}}$ 이고, ω는 $Len(Q[i:j])(=j-i+1)$로서 윈도우 크기에 해당한다. Loh 등은 식 (5)를 사용하여, 정규화 변환된 질의 윈도우 $\overline{Q[i:j]}$가 정규화 변환된 데이터 윈도우 $\overline{S[i:j]}$와 ε'-매치할 때, 해당 데이터 윈도우를 포함하는 서브시퀀스를 후보로 삼는 방법을 사용하였다. 즉, 데이터 시퀀스를 나눈 윈도우를 정규화 변환하여 다차원 색인에 저장하고, 이 다차원 색인을 검색할 때, 주어진 허용치인 ε 대신 ε'을 사용하는 방법을 제안하였다.

3. 연구 동기

Loh 등의 방법은 질의 시퀀스 길이가 색인 구성에 사용된 윈도우 크기보다 두 배 이상 커지는 경우 성능이 크게 저하되는 문제점이 있다. 이는 질의 시퀀스의 길이가 커질수록 색인의 검색 범위인 ε'이 FRM에서 사용하는 검색 범위인 $\varepsilon/\sqrt{p}$에 비해 매우 커지기 때문이다. (그림 1)은 랜덤



(그림 1) 랜덤 워크 데이터 집합에서의 $\varepsilon/\sqrt{p}$에 대한 ε' 값의 변화 (ω =256)

워크 데이터 집합(random walk data set)[3]에 대해, 윈도우 크기를 256으로 고정하고 질의 시퀀스 길이를 256, 512, 768, 1024로 달리 했을 때의 $\varepsilon/\sqrt{p}$에 대한 ε' 값의 변화, 즉 $\frac{\varepsilon'}{\varepsilon/\sqrt{p}}$의 변화를 보여준다. 그림에서 보듯이, 윈도우 크기보다 질의 시퀀스 길이가 커질수록 검색 범위인 ε'의 값이 $\varepsilon/\sqrt{p}$에 비해 매우 커짐을 알 수 있다. 즉, 질의 시퀀스의 길이가 윈도우 크기보다 큰 경우, 색인 검색 범위인 ε' 값이 커져서 색인 페이지 액세스가 많아질 뿐 아니라 후보 개수 증가로 인해 후처리 과정의 데이터 페이지 액세스가 많아지게 된다. 이러한 문제점을 해결하기 위하여, Loh 등은 여러 개의 인덱스를 사용하는 인덱스 보간법을 제안하였다. 인덱스 보간법이란 여러 크기의 윈도우에 대해서 여러 개의 다차원 색인을 구성하고, 질의 시퀀스 길이에 따라 적절한 다차원 색인을 선택하여 서브시퀀스 매칭을 수행하는 방법이다.

그런데, 인덱스 보간법을 사용할 경우, 색인 저장 공간의 오버헤드와 여러 색인에 따른 관리 오버헤드가 발생한다. 정규화 변환을 지원하지 않는 기존 서브시퀀스 매칭 방법[6, 10, 11]에서는 질의 시퀀스 길이가 윈도우 크기의 2~4배가 되더라도 하나의 색인으로 처리가 가능하다. 반면에, Loh 등의 방법은 정규화 변환을 지원하나, 성능 저하를 막기 위해서 인덱스 보간법에 기반한 여러 개(윈도우 크기가 256인 경우, 길이 256~1024인 질의 시퀀스 처리를 위해 아홉 개의 색인을 사용함[9])의 색인을 생성하여야 한다. 이렇게 되면 우선 다차원 색인을 여러 개 생성하기 위한 저장 공간의 오버헤드가 발생하는 단점이 있다. 또한, 원 데이터의 변경 시, 이를 바탕으로 생성된 여러 개의 다차원 색인을 동시에 변경해야 하는 관리 오버헤드가 발생하는 단점이 있다. 이외에도, 색인 검색 범위인 ε'을 구할 때, 제곱근 내의 값이 음수가 되는 경우($\omega^2-\omega\cdot\varepsilon^2\cdot\frac{\sigma^2(Q)}{\sigma^2(Q[i:j])}<0$), 색인을 사용하지

1) 시퀀스 S의 윈도우 s_i를 정규화 변환한 윈도우가 $\bar{S}$의 i번째 디스조인트 윈도우인 $\bar{s}_i$가 아님에 유의한다. 즉, $\bar{s}_i$는 $\bar{S}[(i-1)\cdot\omega+1:i\cdot\omega]$이며, s_i를 정규화 변환한 윈도우는 $\overline{S[(i-1)\cdot\omega+1:i\cdot\omega]}$이다.

2) $\overline{S[i:j]}$가 $\bar{S}[i:j]$와 다름에 유의한다. 즉, $\overline{S[i:j]}$는 S의 서브시퀀스 S[i:j]를 정규화 변환한 것이고, $\bar{S}[i:j]$는 S를 정규화 변환한 시퀀스 $\bar{S}$의 서브시퀀스이다.

3) 시작 엔트리를 1.5로 하고, 각 엔트리에 (-0.001,0.001) 사이의 임의의 값 하나를 더하여 다음 엔트리를 구하는 방식으로 생성된 데이터 시퀀스이다[7, 11, 12].

못하고 순차 스캔을 사용해야 하는 문제점이 있다. 실험 결과, 질의 시퀀스 길이가 윈도우 크기의 두 배 이상이 되면, 색인을 사용하지 못하고 순차 스캔을 해야 하는 경우가 종종 발생하는 것으로 나타났다.

본 연구의 동기는 Loh 등이 사용하지 못한 보조정리 1과 2를 정규화 변환 서브시퀀스 매칭에 그대로 사용하여 색인 검색 범위를 줄이자는데 있다. 즉, 보조정리 1과 2를 정규화 변환된 시퀀스에 적용한 다음 식 (6)과 (7)을 사용하자는 데 본 연구의 동기가 있다. 다음 식 (6)과 (7)을 사용하게 되면, FRM에서와 마찬가지로 색인 검색 범위가 ε에서 $\varepsilon/\sqrt{p}$ 으로 줄어들어 효과적인 서브시퀀스 매칭을 수행할 수 있다.

$$D(\bar{S},\bar{Q}) \le \varepsilon \Rightarrow \bigvee_{i=1}^{p} D(\bar{s}_i,\bar{q}_i) \le \varepsilon/\sqrt{p} \quad (6)$$

$$D(\bar{S},\bar{Q}) \le \varepsilon \Rightarrow D(\bar{S}[i:j],\bar{Q}[i:j]) \le \varepsilon \quad (7)$$

그런데, 식 (6)과 (7)을 정규화 변환 서브시퀀스 매칭에 적용하기 위해서는, 각각의 데이터 윈도우에 대해서 해당 윈도우를 포함하는 다양한 길이 및 다양한 위치의 서브시퀀스로 정규화 변환하는 작업이 필요하다. 즉, 데이터 시퀀스 S를 나눈 윈도우 $S[a:b]$에 대해서, $S[a:b]$를 포함하는 모든 가능한 서브시퀀스 $S[i:j]$를 사용하여 정규화 변환하고 관리하는 과정이 필요하다. 따라서, 본 논문에서는 윈도우가 주어졌을 때, 해당 윈도우를 포함하는 서브시퀀스로 정규화하는 포함-정규화 변환을 정의하고, 이를 사용하여 정규화 변환 서브시퀀스 매칭을 수행하는 방법을 제안한다.

4. 단일 색인을 사용한 정규화 변환 서브시퀀스 매칭

본 장에서는 단일 색인을 사용한 정규화 변환 서브시퀀스 매칭을 제안한다. 제4.1절에서는 제안하는 매칭 방법의 개념을 설명하고, 제4.2에서는 제안하는 개념을 기존 서브시퀀스 매칭인 FRM에 적용한 정규화 변환 서브시퀀스 매칭 방법을 설명한다.

4.1 개념

먼저, 서론에서 제시한 포함-정규화 변환을 다음과 같이 정형적으로 정의한다.

[정의 1] 시퀀스 S의 서브시퀀스 $S[a:b]$를 서브시퀀스 $S[i:j](i \le a < b \le j)$로 **포함-정규화 변환**한 시퀀스 $\overline{S_{[i:j]}[a:b]}$의 엔트리 $\overline{S_{[i:j]}[k]}\ (a \le k \le b)$는 $\dfrac{S[k]-\mu(S[i:j])}{\sigma(S[i:j])}$로 정의한다. □

정의 1에 따르면, 포함-정규화 변환은 서브시퀀스 $S[a:b]$를 $S[a:b]$의 평균과 표준편차가 아닌 $S[a:b]$를 포함하는 $S[i:j]$의 평균과 표준편차로 정규화하는 변환이다. 이러한 개념은 서브시퀀스 매칭에서 주로 사용하는 시계열 데이터의 경우, 이웃한 엔트리의 값의 변화가 크지 않다는 관찰에 기반한다. 즉, 시퀀스 S를 나눈 윈도우 하나를 $S[a:b]$이라 하고, $S[a:b]$를 포함하는 서브시퀀스를 $S[i:j]$라 했을 때, $S[a:b]$의 모든 엔트리들은 $S[i:j]$에 포함되므로, $\mu(S[i:j])$와 $\mu(S[a:b])$, $\sigma(S[i:j])$와 $\sigma(S[a:b])$는 각각 비슷한 값을 가질 것이다. 따라서, $S[a:b]$를 정규화 변환한 시퀀스 $\overline{S[a:b]}$와 $S[a:b]$를 $\mu(S[i:j])$와 $\sigma(S[i:j])$를 사용하여 포함-정규화 변환한 시퀀스 $\overline{S_{[i:j]}[a:b]}$는 유사한 값을 가질 것이다. 결과적으로, 윈도우 $S[a:b]$를 여러 서브시퀀스 $S[i:j]$들로 포함-정규화 변환한 윈도우 $\overline{S_{[i:j]}[a:b]}$들은 유사한 엔트리들을 가질 것이고, 본 논문에서는 이러한 성질을 사용하여 새로운 정규화 변환 서브시퀀스 매칭 방법을 제안한다.

제안하는 정규화 변환 서브시퀀스 매칭에서는 데이터 시퀀스를 나눈 각 윈도우를 다차원 색인의 하나의 MBR로 매핑하는 방법을 사용한다. FRM에서는 데이터 시퀀스를 나눈 슬라이딩 윈도우를 하나의 점으로 매핑하는 방법을 사용하였다. 반면에, 제안하는 방법에서는 각 윈도우를 하나의 점이 아닌 여러 개의 점을 포함하는 MBR로 매핑시킨다. 즉, 데이터 시퀀스를 나눈 윈도우를 직접 정규화 변환하는 대신, 해당 윈도우를 포함하는 여러 서브시퀀스를 사용하여 여러 개의 윈도우로 포함-정규화 변환하고, 이들 윈도우를 저차원 변환한 점들을 포함하는 MBR을 구성한다. 결국, 하나의 윈도우는 하나의 점이 아닌 하나의 MBR로 변환되며, 이 MBR을 다차원 색인에 저장한다. 그런데, 앞서 설명한 바와 같이, 포함-정규화 변환된 윈도우들은 각 엔트리의 값이 매우 유사할 것이므로, 다차원 색인에 저장될 MBR의 크기는 그다지 크지 않게 된다.

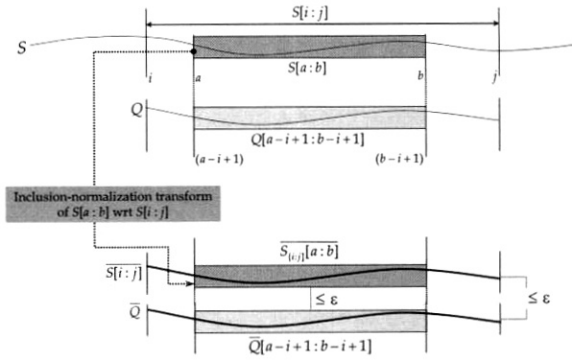
제안하는 정규화 변환 서브시퀀스 매칭을 설명하기 위하여, 질의 시퀀스와 비교 대상이 되는 서브시퀀스가 1) 윈도우를 하나 포함하는 경우와 2) 윈도우를 두 개 이상 포함하는 경우로 구분하여 설명한다. 우선, 윈도우를 하나 포함하는 경우에 대해서는 다음 보조정리 3이 성립한다.

[보조정리 3] 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$에 윈도우 $S[a:b]$가 포함되어 있고, $S[i:j]$와 비교 대상이 되는 질의 시퀀스를 Q라 하면, 다음 식 (8)이 성립한다.

$$D(\overline{S[i:j]},\bar{Q}) \le \varepsilon \Rightarrow D(\overline{S_{[i:j]}[a:b]},\bar{Q}[a-i+1:b-i+1]) \le \varepsilon \text{, where } i \le a < b \le j \quad (8)$$

[증명] 부록 A 참조 □

(그림 2)는 보조정리 3을 설명한 것이다. (그림 2)를 보면, 정규화 변환된 두 시퀀스 $\overline{S[i:j]}$와 $\bar{Q}$가 ε-매치할 경우, $\overline{S[i:j]}$에 포함된 윈도우 $\overline{S_{[i:j]}[a:b]}$와 이에 대응하는 $\bar{Q}$에 포함된 윈도우 $\bar{Q}[a-i+1:b-i+1]$가 ε-매치한다. 다시 말해서, 만일 $\overline{S[i:j]}$와 $\bar{Q}$가 ε-매치하면, $\overline{S[i:j]}$에 포함된 윈도



(그림 2) 서브시퀀스에 윈도우가 하나 포함된 경우 포함-정규화 변환의 사용.

우 $\overline{S_{[i:j]}[a:b]}$와 $\overline{Q}$에 포함된 윈도우 $\overline{Q}[a-i+1:b-i+1]$이 반드시 ε-매치한다. 따라서, 포함-정규화 변환된 윈도우 $\overline{S_{[i:j]}[a:b]}$와 $\overline{Q}[a-i+1:b-i+1]$이 ε-매치하는 경우, 즉, 식 (8)의 필요조건이 만족하는 경우, $\overline{S[i:j]}$를 $\overline{Q}$와 ε-매치하는 후보로 삼으면 착오기각이 발생하지 않게 된다.

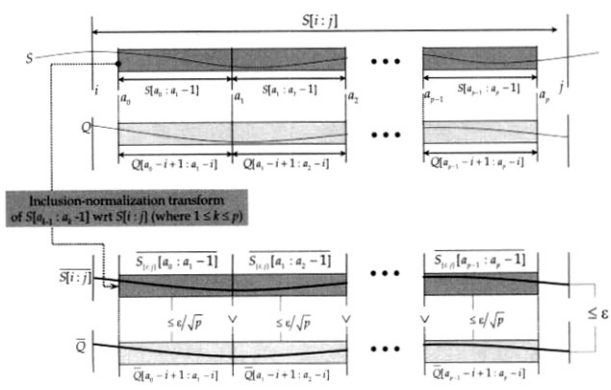
다음으로, 윈도우를 두 개 이상 포함하는 경우에 대해서는 다음 보조정리 4가 성립한다.

[보조정리 4] 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$에 크기 ω인 p개의 윈도우 $S[a_0:a_1-1]$, $S[a_1:a_2-1]$, ..., $S[a_{p-1}:a_p-1]$가 포함되어 있고, $S[i:j]$와 비교 대상이 되는 질의 시퀀스를 Q라 하면, 다음 식 (9)가 성립한다.

$$D(\overline{S[i:j]},\overline{Q})\le\varepsilon \Rightarrow \bigvee_{k=1}^{p} D(\overline{S_{[i:j]}[a_{k-1}:a_k-1]},\overline{Q}[a_{k-1}-i+1:a_k-i])\le\varepsilon/\sqrt{p}\ ,\quad \text{where } i\le a_{k-1}<a_k\le j+1,\ 1\le k\le p,\ \omega=a_k-a_{k-1} \tag{9}$$

[증명] 부록 B 참조 □

(그림 3)은 보조정리 4를 설명한 것이다. (그림 3)을 보면, 두 시퀀스 $\overline{S[i:j]}$와 $\overline{Q}$가 ε-매치할 경우, $\overline{S[i:j]}$와 $\overline{Q}$에 각각



(그림 3) 서브시퀀스에 윈도우가 두 개 이상 포함된 경우 포함-정규화 변환의 사용.

포함된 p개의 윈도우 쌍 $\overline{S_{[i:j]}[a_{k-1}:a_k-1]}$와 $\overline{Q}[a_{k-1}-i+1:a_k-i]$ 중 최소한 하나가 반드시 $\varepsilon/\sqrt{p}$-매치한다. 따라서, 윈도우 쌍 $\overline{S_{[i:j]}[a_{k-1}:a_k-1]}$와 $\overline{Q}[a_{k-1}-i+1:a_k-i]$중 최소한 하나가 $\varepsilon/\sqrt{p}$-매치하는 경우, 즉, 식 (9)의 필요조건이 만족하는 경우, $\overline{S[i:j]}$를 $\overline{Q}$와 ε-매치하는 후보 서브시퀀스로 삼으면 착오기각이 발생하지 않게 된다.

제안하는 정규화 변환 서브시퀀스 매칭 방법은 FRM의 서브시퀀스 매칭 방법에 포함-정규화 변환을 적용한 방법이다(이를 간략히 **FRM-NT**(FRM that supports Normalization Transform)이라 한다)[4]. 제안하는 FRM-NT는 다음 정리 1에 기반하여 정규화 변환 서브시퀀스 매칭을 수행한다.

[정리 1] 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$를 정규화 변환한 $\overline{S[i:j]}$와 질의 시퀀스 Q를 정규화 변환한 $\overline{Q}$가 ε-매치한다면, 적어도 하나 이상의 $\overline{Q}$의 k번째 디스조인트 윈도우 $\overline{q_k}$와 $\overline{S[i:j]}$에 포함된 슬라이딩 윈도우 $\overline{S_{[i:j]}[i+(k-1)\cdot\omega:i+k\cdot\omega-1]}$가 $\varepsilon/\sqrt{p}$-매치한다. 여기에서, $1\le k\le p$ 이고, $p=\lfloor Len(Q)/\omega\rfloor$이다. 즉, 다음 식 (10)이 성립한다.

$$D(\overline{S[i:j]},\overline{Q})\le\varepsilon \Rightarrow \bigvee_{k=1}^{p} D(\overline{S_{[i:j]}[i+(k-1)\cdot\omega:i+k\cdot\omega-1]},\overline{q_k})\le\varepsilon/\sqrt{p} \tag{10}$$

[증명] 우선, 보조정리 4에 의하여 다음 식 (11)이 성립한다.

$$D(\overline{S[i:j]},\overline{Q})\le\varepsilon \Rightarrow \bigvee_{k=1}^{p} D(\overline{S_{[i:j]}[a_{k-1}:a_k-1]},\overline{Q}[a_{k-1}-i+1:a_k-i])\le\varepsilon/\sqrt{p} \tag{11}$$

여기에서, $a_0=i$, $a_1=i+\omega$, ..., $a_{p-1}=i+(p-1)\cdot\omega$라 하자. 즉, $a_{k-1}=i+(k-1)\cdot\omega$ $(1\le k\le p)$라 하자. 그러면, 식 (11)은 다음 식 (12)와 같이 표현할 수 있다.

$$D(\overline{S[i:j]},\overline{Q})\le\varepsilon \Rightarrow \bigvee_{k=1}^{p} D(\overline{S_{[i:j]}[i+(k-1)\cdot\omega:i+k\cdot\omega-1]},\overline{Q}[(k-1)\cdot\omega+1:k\cdot\omega])\le\varepsilon/\sqrt{p} \tag{12}$$

그런데, <표 1>의 표기법에 의해서, $\overline{Q}[(k-1)\cdot\omega+1:k\cdot\omega])=\overline{q_k}$ 이므로, 식 (12)는 다시 다음 식 (13)으로 표현되며, 이에 따라 증명이 완료된다.

4) 포함-정규화 변환을 DualMatch[11]나 GeneralMatch[12]에 적용하는 방법을 생각할 수도 있다. 그러나, 이들 방법에 포함-정규화 변환을 적용하는 경우, 색인에 저장해야 하는 점(혹은 MBR)의 개수가 FRM과 유사하여 성능에 큰 차이가 없는 것으로 분석되었다. 따라서, 본 논문에서는 포함-정규화 변환을 가장 쉽게 적용할 수 있는 FRM만을 다룬다.

$$D(\overline{S[i:j]},\overline{Q}) \le \varepsilon \Rightarrow$$

$$\bigvee_{k=1}^{p} D(\overline{S_{[i:j]}[i+(k-1)\cdot\omega : i+k\cdot\omega-1]},\overline{q_k}) \le \varepsilon/\sqrt{p} \qquad (13)\ \square$$

결국, 정리 1에 따라서, 디스조인트 윈도우 $\overline{q_k}$와 슬라이딩 윈도우 $\overline{S_{[i:j]}[i+(k-1)\cdot\omega : i+k\cdot\omega-1]}$가 $\varepsilon/\sqrt{p}$-매치할 때, 즉, 식 (10)의 필요조건이 만족할 때, $\overline{S[i:j]}$를 $\overline{Q}$와 ε-매치하는 후보 서브시퀀스로 삼으면 착오기각이 발생하지 않는다. 여기에서, $a = i+(k-1)\cdot\omega$라 표현하면, $i = a-(k-1)\cdot\omega$이 된다. 다시 말해서, FRM-NT에서는 윈도우 $\overline{S_{[i:j]}[a:a+\omega-1]}$과 $\overline{q_k}$가 $\varepsilon/\sqrt{p}$-매치할 때, 시작 위치를 i $(=a-(k-1)\cdot\omega)$로 하는 서브시퀀스 $\overline{S[i:j]}$를 $\overline{Q}$와 ε-매치하는 후보 서브시퀀스로 삼으면 착오기각이 발생하지 않는다.

4.2 단일 색인 기반의 정규화 변환 서브시퀀스 매칭

본 절에서는 FRM-NT의 색인 구성 알고리즘과 서브시퀀스 매칭 알고리즘을 설명한다. 우선, FRM-NT의 색인 구성 알고리즘은 (그림 4)와 같다. 알고리즘의 스텝 (1)에서는 데이터 시퀀스 S를 크기 ω의 슬라이딩 윈도우로 나눈다. 다음으로, 스텝 (2)~(6)에서는 각 슬라이딩 윈도우를 다차원 색인의 하나의 MBR로 저장하는 작업을 수행한다. 먼저, 스텝 (3)에서는 각 슬라이딩 윈도우를 가능한 모든 서브시퀀스들을 대상으로 포함 정규화 변환하여, 변환된 윈도우 집합을 구성한다. 이때의 서브시퀀스 길이는 가능한 모든 길의 시퀀스 길이 각각이 되며, 서브시퀀스의 시작 위치는 해당 윈도우를 포함하는 모든 위치 각각이 된다. 다음으로, 스텝 (4)에서는 변환된 윈도우들을 저차원 변환하여 f-차원 MBR을 구성한다. 마지막으로, 스텝 (5)에서는 구성한 MBR을 해당 윈도우의 시작 위치와 함께 다차원 색인에 저장한다. 이와 같은 과정을 거쳐 생성한 다차원 색인은 이후 서브시퀀스 매칭 알고리즘에 사용된다.

FRM-NT의 색인 구성 알고리즘을 보면, 서브시퀀스가 생성될 수 있는 위치 각각에 대해서 포함-정규화 변환된 윈도우들을 구성한다((그림 4)의 스텝 (3) 참조). (그림 6)은 이와 같이 여러 위치에 대해 포함 정규화 변환을 수행하는 배경을 설명한다. 즉, (그림 5)는 크기 ω인 슬라이딩 윈도우 $S[a:b]$에 대해서, 길이 q_{len}인 서브시퀀스 $S[i:j]$($j-i+1=q_{len}$)를 구

Procedure *FRM-NT-BuildIndex*(Data Sequence *S*, Window size ω)

(1) Divide *S* into sliding windows of length ω;

(2) **for** each sliding window $S[a:b]$ **do**

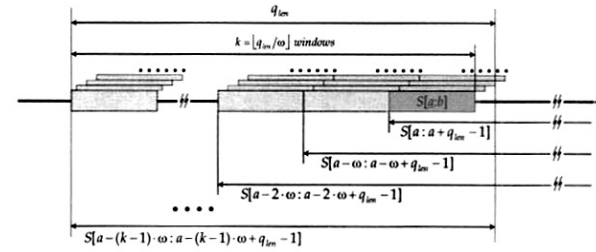
(3) Make a set of inclusion-normalization transformed windows $\overline{S_{[i:j]}[a:b]}$ for each possible query length and for each possible position of $S[i:j]$;

(4) Construct an *f*-dimensional MBR *f-D MBR* by using lower-dimensional transformations on a set of $\overline{S_{[i:j]}[a:b]}$'s;

(5) Make a record <*f-D MBR*, *offset*=*a*>, and store it into the index;

(6) **endfor**

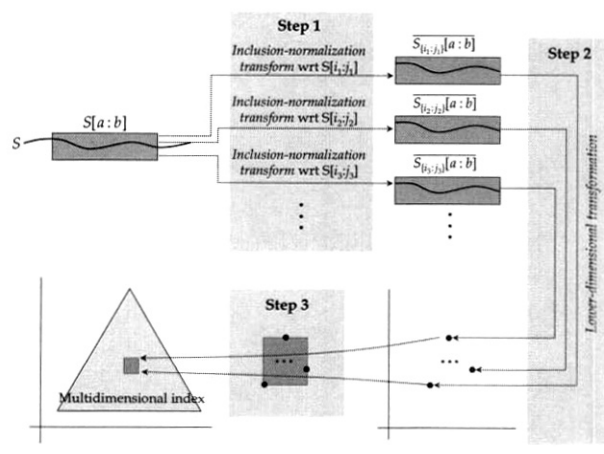
(그림 4) FRM-NT의 색인 구성 알고리즘



(그림 5) 윈도우 S[a:b]를 포함하는 길이 q_{len}인 서브시퀀스 S[i:j]의 가능한 위치

성하는 위치를 나타낸다. 그림에서 보면, $S[a:b]$를 포함하는 길이 q_{len}인 서브시퀀스 $S[i:j]$의 시작 위치 i는 a, $a-\omega$, $a-2\omega$, $\cdots$, $a-(\lfloor q_{len}/\omega \rfloor -1)\cdot\omega$로서, $i = a-k\cdot\omega$ $(0 \le k \le \lfloor q_{len}/\omega \rfloor -1)$임을 알 수 있다. 결국, FRM-NT에서는 윈도우 $S[a:b]$를 길이 q_{len}인 각 서브시퀀스 $S[i:j]$($i=a-k\cdot\omega$, $j=i+q_{len}-1$, $0 \le k \le \lfloor q_{len}/\omega \rfloor -1$)로 포함-정규화 변환하여, 변환된 윈도우 $\overline{S_{[i:j]}[a:b]}$들을 구성하는 방법을 사용한다.

(그림 6)은 포함 정규화 변환 및 저차원 변환을 통하여 하나의 윈도우에 대해서 색인에 저장할 MBR이 구성되는 과정을 나타낸다. 그림에서 보면, 단계 1에서는 데이터 시퀀스 S를 나눈 각 윈도우 $S[a:b]$를 가능한 각 서브시퀀스 $S[i_k:j_k]$를 사용하여 포함-정규화 변환한 윈도우 $\overline{S_{[i_k:j_k]}[a:b]}$를 생성한다. 다음으로, 단계 2에서는 이들 변환된 윈도우들을 저차원 공간의 점으로 변환한다. 마지막으로, 단계 3에서는 이들 점을 포함하는 MBR을 구성하고, 이 MBR을 다차원 색인에 저장하여 서브시퀀스 매칭에 사용하게 된다. 이와 같이 포함-정규화 변환을 사용하는 방법이 가능한 이유는 윈도우 $S[a:b]$를 $S[i_1:j_1]$로 포함-정규화 변환한 $\overline{S_{[i_1:j_1]}[a:b]}$와 $S[i_2:j_2]$로 변환한 $\overline{S_{[i_2:j_2]}[a:b]}$가 유사한 값을 가질 것이라는 관찰에 근거한다. 즉, 포함-정규화 변환에 사용된 $S[i_1:j_1]$과 $S[i_2:j_2]$가 다를지라도, 상당수의 엔트리가 포



(그림 6) 포함-정규환 변환을 적용한 각 윈도우의 다차원 색인 저장 과정

Procedure *FRM-NT-SubsequenceMatching* (Query Sequence Q, Window size ω)

(1) Make $\overline{Q}$ from Q by using the normalization transform;
(2) Divide $\overline{Q}$ into disjoint windows $\overline{q_i}(1 \le i \le p)$ of length ω;
(3) **for** each disjoint window $\overline{q_i}$ **do**
(4) Transform the window to an f-dimensional point by using the feature extraction function;
(5) Construct a range query using the point and $\varepsilon/\sqrt{p}$;
(6) Search the index and find the records of the form <*f-D MBR, offset*>;
(7) Include in the candidate set the subsequences $S[i:j]$ for each possible position;
// where $i = (offset + l) - k \cdot \omega$, $j = i + Len(Q) - 1$, $0 \le l \le \omega - 1$, and $0 \le k \le \lfloor q_{len}/\omega \rfloor - 1$
(8) **endfor**
(9) Do the post-processing step;

(그림 7) FRM-NT의 서브시퀀스 매칭 알고리즘

함-정규화 과정에 공통으로 사용되기 때문에, 변환된 두 윈도우의 엔트리 값은 유사하다는 관찰에 근거한다.

그런데, FRM-NT는 데이터 시퀀스를 슬라이딩 윈도우로 나누므로, 너무 많은 윈도우가 생성되는 문제가 있다. 이러한 문제점을 해결하기 위하여, FRM에서는 여러 개의 슬라이딩 윈도우를 하나의 MBR에 포함시키는 방법을 사용하였다[6]. 따라서, 제안하는 FRM-NT에서도 여러 개의 슬라이딩 윈도우가 변환된 여러 개의 MBR을 하나의 MBR에 포함시키는 방법을 사용한다. 즉, 연속된 여러 슬라이딩 윈도우를 대표하는 MBR을 구성하고, 이 MBR의 첫 번째 및 마지막 슬라이딩 윈도우 시작 위치와 함께 다차원 색인에 저장하는 방법을 사용한다. 그러나, 설명의 편의상, 본 논문에서는 (그림 4)의 알고리즘과 같이 각 슬라이딩 윈도우에 해당하는 MBR을 직접 저장하고 서브시퀀스 매칭을 수행하는 것으로 기술한다.

다음으로, 다차원 색인이 구성된 후에는 (그림 7)의 알고리즘을 사용하여 정규화 변환 서브시퀀스 매칭을 수행한다. 알고리즘의 스텝 (1)과 (2)에서는 질의 시퀀스 Q를 정규화 변환한 후, 디스조인트 윈도우 $\overline{q_i}$ 들을 구성한다. 다음으로, 라인 (3)에서 (8)까지는 각 디스조인트 윈도우 $\overline{q_i}$ 에 대해서, 허용치 $\varepsilon/\sqrt{p}$ 으로 다차원 색인을 검색하여, $\overline{q_i}$ 와 $\varepsilon/\sqrt{p}$ -매치하는 MBR(<*f-D MBR, offset*>)에 대해서 서브시퀀스 $S[i:j]$들을 후보 서브시퀀스로 삼는다. 마지막으로, 라인 (9)에서는 이들 후보 서브시퀀스들에 대해서 후처리 과정[1, 6, 10]을 진행하여, 착오해답을 제거하고 유사 서브시퀀스만을 찾는다.

5. 성능 평가

본 장에서는 제안한 FRM-NT의 성능 평가 결과를 설명한다. 제4.1절에서는 성능 평가를 수행한 실험 데이터와 실험 환경을 설명하고, 제4.2절에서는 실험 결과를 설명한다.

5.1 실험 환경

제안한 방법의 우수성을 입증하기 위하여 두 가지 종류의 데이터를 사용하여 실험을 수행하였다. 사용한 데이터는 하나의 긴 데이터 시퀀스로 구성된 것으로서, 이는 여러 개의 데이터 시퀀스로 구성된 경우와 동일한 효과를 가진다[6, 11]. 첫 번째 데이터는 기존 연구[6, 11, 12]에서 사용한 실제 주식 데이터로서 약 33만개의 엔트리로 구성되어 있으며, 이를 STOCK-DATA라 한다. 두 번째 데이터는 합성 데이터(synthetic data)로서 데이터 시퀀스의 시작 엔트리를 1.5로 하고, 각 엔트리에 (-0.001, 0.001) 사이의 임의의 값 하나를 더하여 다음 엔트리를 구하는 방식으로 생성된 100만개의 랜덤 워크 데이터(random walk data)이다. 이 데이터 역시 기존 연구[6, 11, 12]에서 사용한 것으로서 이를 WALK-DATA라 한다.

성능 평가는 제안한 FRM-NT와 Loh 등[9]의 기존 연구(저자의 이름을 따서 간략히 **LKW**라 한다)를 대상으로 하였다. 실험을 수행한 하드웨어 플랫폼은 Intel Pentium IV 2.80 GHz CPU, 512MB RAM, 70GB 하드 디스크를 장착한 PC이며, 소프트웨어 플랫폼은 GNU/Linux Version 2.6.6 운영체제이다. 다차원 색인으로는 두 방법 모두 R*-트리[3]를 사용하였으며, 데이터 페이지 및 색인 페이지의 크기는 4096 바이트를 사용하였다. 첫 번째 실험에서는, 공평한 비교를 위하여 두 방법 모두 하나의 색인을 생성하고 실험하였다. 이 실험에서는, 최소 질의 시퀀스 길이로 256을 사용하여, 두 방법 모두 윈도우 크기는 최소 질의 시퀀스 길이와 같은 256으로 설정하였고, 가능한 질의 시퀀스 길이는 256, 512, 768, 1024의 네 가지를 사용하였다. 두 번째 실험에서는, 여러 색인을 사용하는 색인 보간법 측면에서 두 가지 방법을 비교하였다. 이를 위해, 윈도우 크기 256, 512, 1024에 대해서 색인을 구성하고, 이들 길이를 포함하여 384, 640, 768, 896에 대해 질의 시퀀스를 구성하고 성능 실험을 수행하였다. 저차원 변환을 위한 특성 추출 함수로는 DFT(Discrete Fourier Transform) 변환[12]과 웨이블릿(Wavelet) 변환[4]을 사용하였으며, 특성은 6개를 사용하였다[6, 11].

실험 결과로는 두 방법의 실제 수행 시간을 측정하였다. 질의 시퀀스는 데이터 시퀀스의 임의 위치(random offset)를 시작 엔트리로 하는 $Len(Q)$의 서브시퀀스를 추출하여 사용하였으며, 노이즈(noise) 효과를 피하기 위하여 같은 길이를 갖는 10개의 다른 질의 시퀀스에 대해서 실험한 후 평균을 취한 값을 실험 결과로 하였다. 질의에 대한 선택률[6,9]은 10^{-5}으로 설정하였는데, 이는 대용량 시계열 데이터에서 일부 결과만을 선택하는 경우, 즉 선택률이 낮은 경우가 실용적으로 의미를 가지기 때문이다[9, 11, 12].

5.2 실험 결과

본 절에서는 세 가지 방법의 실험 결과를 설명한다. 먼저, 실험 1)에서는 하나의 색인을 사용하고 특성 추출 함수로 DFT 변환을 사용하여 STOCK-DATA와 WALK-DATA를 대상으로 실험한 결과를 설명한다. 다음으로, 실험 2)에서는 하나의 색인을 사용하고 특성 추출 함수로 웨이블릿 변환을 사용한 경우의 실험 결과이다. 마지막으로, 실험 3)에서는 복수 개의 색인을 사용하는 경우의 실험 결과를 설명한다.

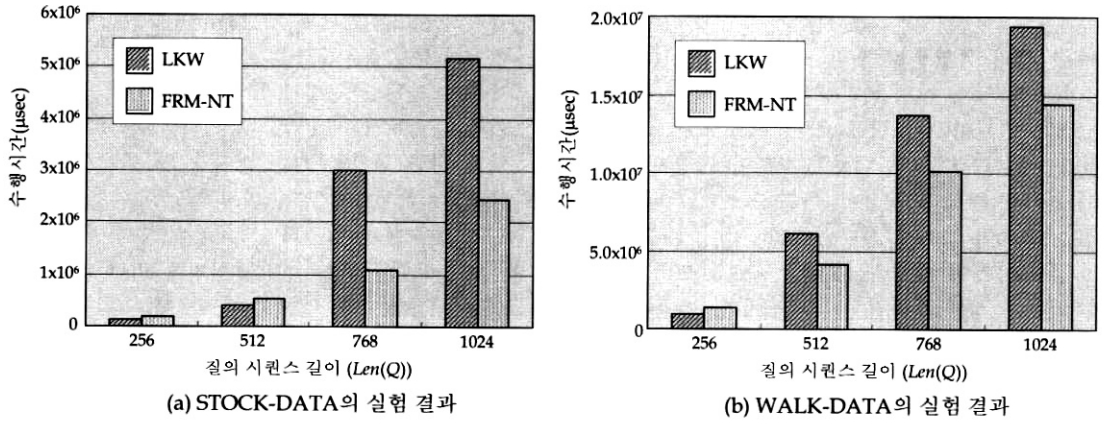
실험 1) 단일 색인을 사용하고, DFT 변환을 사용한 경우의 실험 결과

(그림 8)은 하나의 색인을 사용하고 DFT 변환을 사용한 경우 두 방법의 수행 시간을 나타낸다. (그림 8)의 (a)는 STOCK-DATA에 대한 실험 결과이고, (b)는 WALK-DATA에 대한 실험 결과이다. 그림을 보면, 질의 시퀀스의 길이가 256인 경우에는 기존 방법인 LKW가 제안한 FRM-NT에 비해 성능이 약간 우수함을 알 수 있다. 이는 질의 시퀀스 길이와 윈도우 크기가 같은 경우, LKW는 ε'이 ε과 같아 최적의 검색 조건을 제공하기 때문이다. 반면에, FRM-NT는 포함-정규화 변환에 의한 점 대신 MBR 구성으로 인하여, 색인에 저장되는 MBR 크기가 LKW보다 크기 때문이다. 즉, 질의 시퀀스 길이와 윈도우 크기가 같은 경우, 두 방법 모두 검색 범위는 ε으로 같은 반면에, 색인에 저장되는 MBR의 크기는 FRM-NT가 LKW보다 크기 때문에, LKW가 FRM-NT보다 좋은 성능을 나타내는 것이다. 그러나, 질의 시퀀스가 256보다 큰 경우, 즉 질의 시퀀스 길이가 512, 768, 1024인 경우에는 제안하는 FRM-NT가 LKW보다 좋은 성능을 보이고 있다. 제3장에서 설명한 바와 같이, 이는 질의 시퀀스 길이가 커짐에 따라 LKW의 색인 검색 범위는 ε'은 ε보다 커지는 반면에 FRM-NT는 $\varepsilon/\sqrt{p}$으로 검색 범위를 줄일 수 있기 때문이다.

결과적으로 볼 때, LKW는 특정 질의 시퀀스에 최적화한 방법인 반면, FRM-NT는 모든 질의 시퀀스 길이에 대해 골고루 성능을 향상시킨 방법이라 할 수 있다. 즉, LKW는 질의 시퀀스 길이가 윈도우 크기와 동일한 경우에 가장 좋은 성능을 보이나, 질의 시퀀스 길이가 윈도우 크기보다 큰 경우에는 성능이 좋지 않은 결과를 보인다. 반면에, FRM-NT는 질의 시퀀스 길이가 윈도우 크기와 같은 경우에는 LKW에 비해 성능이 떨어지나, 같지 않는 경우에도 LKW에 비해 비교적 우수한 성능을 보임을 알 수 있다. 실험 결과를 요약하면, FRM-NT는 LKW에 비해 STOCK-DATA의 경우 최대 2.8배, WALK-DATA의 경우 최대 1.4배까지 성능을 향상 시킨 것으로 나타났다.

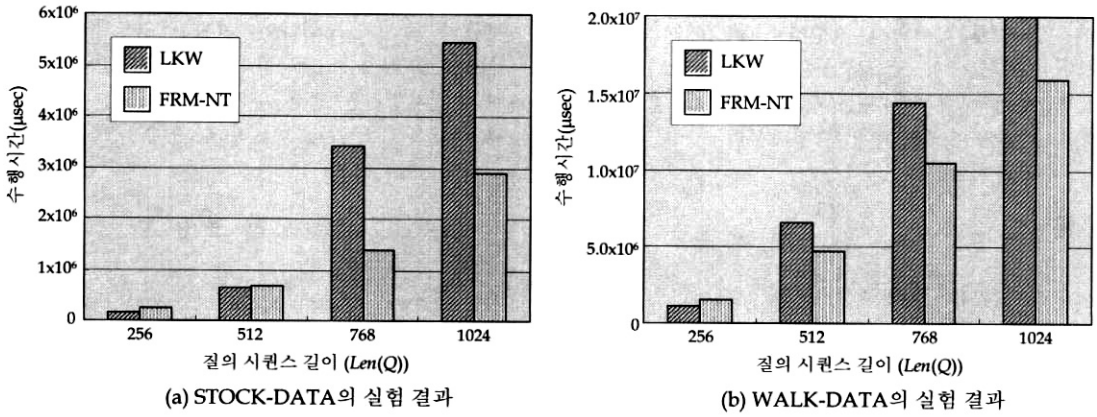
실험 2) 단일 색인을 사용하고 웨이블릿 변환을 사용한 경우의 실험 결과

(그림 9)는 하나의 색인을 사용하고 웨이블릿 변환을 사용한 경우의 두 방법의 수행 시간을 나타낸다. 그림을 보면, 실험 결과가 DFT 변환을 사용한 (그림 8)의 결과와 유사함을 알 수 있다. 즉, 실험 결과는 특성 추출 함수의 종류와 관계없이 유사하게 나타난다고 할 수 있다. (그림 9)의 결과를 보면, (그림 8)과 마찬가지로, 질의 시퀀스의 길이가 256



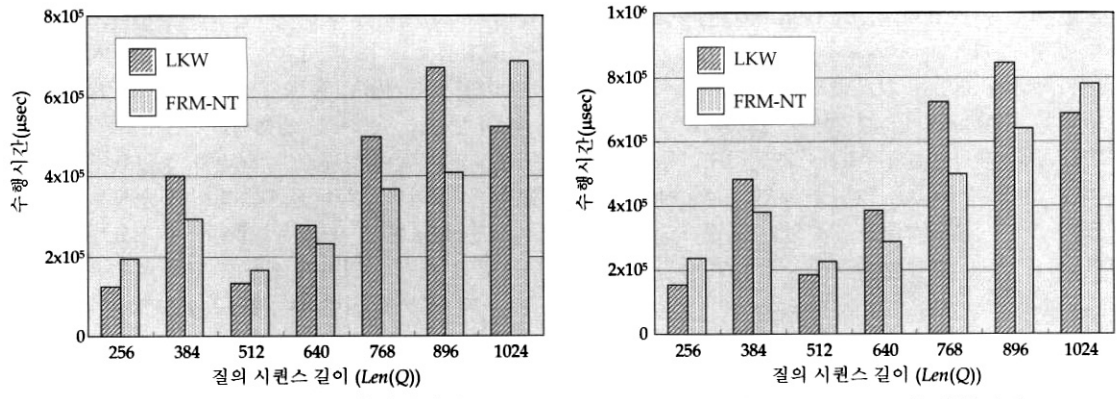
(a) STOCK-DATA의 실험 결과 (b) WALK-DATA의 실험 결과

(그림 8) 단일 색인을 사용하고 DFT 변환을 사용한 경우의 실험 결과



(a) STOCK-DATA의 실험 결과 (b) WALK-DATA의 실험 결과

(그림 9) 단일 색인을 사용하고 웨이블릿 변환을 사용한 경우의 실험 결과



(a) STOCK-DATA의 실험 결과 (b) WALK-DATA의 실험 결과

(그림 10) 복수 개 색인을 사용하고 DFT 변환을 사용한 경우의 실험 결과

인 경우에는 LKW의 성능이 FRM-NT 보다 우수함을 알 수 있다. 반면에 질의 시퀀스 길이가 윈도우 크기보다 큰 경우에는 FRM-NT의 성능이 LKW보다 우수한 것으로 나타났다. 앞서 설명한 바와 같이, 이는 LKW의 경우 질의 시퀀스 길이가 윈도우 크기와 같은 경우로 최적화된 반면에, FRM-NT는 다양한 길이를 고루 고려하기 때문이다. 웨이블릿 변환의 실험 결과를 요약하면, STOCK-DATA와 WALK-DATA에 있어서, FRM-NT는 LKW에 비해 수행 시간을 최대 2.5배 및 1.4배까지 각각 줄인 것으로 나타났다.

실험 3) 복수 개의 색인을 사용한 경우의 실험 결과

(그림 10)은 복수 개 색인을 사용하고 DFT 변환을 사용한 경우의 실험 결과이다. 그림을 보면, 질의 시퀀스의 길이가 256, 512, 1024인 경우, 즉 다차원 색인이 생성된 경우에는 LKW가 FRM-NT보다 성능이 우수한 것으로 나타났다. 이는 앞서 실험 1)과 2)에서 설명한 바와 같이, LKW는 질의 시퀀스 길이와 색인 구성에 사용된 윈도우 크기가 동일한 경우에 최적의 성능을 나타내기 때문이다. 반면에, 질의 시퀀스 길이가 384, 640, 768, 896인 경우, 즉 다차원 색인이 생성되지 않은 경우에는 제안한 FRM-NT가 LKW보다 우수한 성능을 보임을 알 수 있다. 이는 FRM-NT의 경우 색인이 생성된 길이뿐 아니라 모든 질의 시퀀스 길이에 대해서 골고루 성능을 향상시키기 때문이다. 이와 같이, 제안한 FRM-NT는 복수 개의 색인을 생성하는 환경에서도 LKW와 유사하거나 우수한 성능을 보임을 알 수 있다. DFT 변환 대신에 Wavelet 변환을 사용한 경우도 유사한 결과를 얻었으며, 지면 관계상 자세한 내용은 생략한다.

6. 결 론

본 논문에서는 하나의 색인을 사용하여 정규화 변환 서브시퀀스 매칭을 효율적으로 수행하는 새로운 접근법을 제안하였다. 기존의 정규화 변환 서브시퀀스 매칭은 여러 색인의 생성에 따른 색인 공간의 오버헤드와 색인 관리의 오버헤드가 발생하는 문제점이 있다. 또한, 질의 시퀀스 길이가 윈도우 크기 보다 큰 일부 경우에 대해서는 색인을 사용하지 못하고 순차 스캔을 해야 하는 경우도 발생한다. 이를 해결하기 위하여, 본 논문에서 우선 정규화 변환을 일반화하여 포함-정규화 변환 개념을 제안하였다. 그리고, 포함-정규화 변환을 사용하면, 하나의 색인을 사용해서도 다양한 길이의 질의 시퀀스에 대해 효율적인 정규화 변환 서브시퀀스 매칭을 수행할 수 있음을 보였다.

본 논문의 공헌은 다음과 같이 요약할 수 있다. 첫째, 기존 정규화 변환 서브시퀀스 매칭의 문제점을 분석하였다. 둘째, 정규화 변환을 일반화하여 포함-정규화 변환 개념을 정형적으로 정의하였다. 셋째, 포함-정규화 변환을 기존 서브시퀀스 매칭 방법인 FRM에 적용하는 방안의 이론적 근거를 정리로서 제시하고 증명하였다. 넷째, 이 방안을 구현하기 위한 색인 구성 알고리즘 및 서브시퀀스 매칭 알고리즘을 각각 제안하였다. 마지막으로, 실험을 통해 제안한 방법의 우수성을 입증하였다. 실제 주식 데이터에 대한 실험 결과, 제안한 방법은 기존 방법에 비해 최대 2.5~2.8배까지 성능을 향상 시킨 것으로 나타났다.

본 논문에서 제안한 정규화 변환 지원 서브시퀀스 매칭은 스케일링 및 쉬프팅, 이동 평균 등 다른 변환을 지원하는 서브시퀀스 매칭으로 일반화 될 수 있다. 즉, 기존의 방법들이 새로운 검색 범위를 찾거나 여러 색인을 사용하여 문제를 해결한 반면에, 제안한 방법은 각 윈도우에 대해서 확장 변환의 개념을 도입하여, 변환된 윈도우를 만들고 이를 서브시퀀스 매칭에 활용한다. 따라서, 제안한 방법은 정규화 변환을 포함하는 많은 다른 종류의 변환을 지원하는 서브시퀀스 매칭에 폭넓게 적용될 수 있을 것으로 사료된다.

참 고 문 헌

[1] Agrawal, R., Faloutsos, C., and Swami, A., "Efficient Similarity Search in Sequence Databases," In Proc. the 4th Int'l Conf. on Foundations of Data Organization and Algorithms, Chicago, Illinois, pp.69-84, Oct., 1993.

[2] Agrawal, R., Lin, K.-I., Sawhney, H. S., and Shim, K., "Fast

Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases," In Proc. the 21st Int'l Conf. on Very Large Data Bases, Zurich, Switzerland, pp.490-501, Sept., 1995.

[3] Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B., "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," In Proc. Int'l Conf. on Management of Data, ACM SIGMOD, Atlantic City, New Jersey, pp.322-331, May, 1990.

[4] Chan, K.-P., Fu, A. W. C., and Yu, C. T., "Haar Wavelets for Efficient Similarity Search of Time-Series: With and Without Time Warping," IEEE Trans. on Knowledge and Data Engineering, Vol.15, No.3, pp.686-705, Jan./Feb., 2003.

[5] Chu, K. W. and Wong, M. H., "Fast Time Series Searching with Scaling and Shifting," In Proc. the 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Philadelphia, Pennsylvania, pp.237-248, June, 1999.

[6] Faloutsos, C., Ranganathan, M., and Manolopoulos, Y., "Fast Subsequence Matching in Time-Series Databases," In Proc. Int'l Conf. on Management of Data, ACM SIGMOD, Minneapolis, Minnesota, pp.419-429, May, 1994.

[7] Kim, S.-W., Park, S, and Chu, W. W., "Efficient Processing of Similarity Search Under Time Warping in Sequence Databases: An Index-based Approach," Information Systems, Vol.29, No.5, pp.405-420, July 2004.

[8] Loh, W.-K., Kim, S.-W., and Whang, K.-Y., "Index Interpolation: A Subsequence Matching Algorithm Supporting Moving Average Transform of Arbitrary Order in Time-Series Databases," IEICE Transactions on Information and Systems, Vol.E84-D, No.1, pp.76-86, 2000.

[9] Loh, W.-K., Kim, S.-W., and Whang, K.-Y., "A Subsequence Matching Algorithm that Supports Normalization Transform in Time-Series Databases," Data Mining and Knowledge Discovery, Vol.9, No.1, pp.5-28, July, 2004.

[10] Moon, Y.-S., Whang, K. Y., and Loh, W.-K., "Duality-Based Subsequence Matching in Time-Series Databases," In Proc. the 17th Int'l Conf. on Data Engineering (ICDE), IEEE, Heidelberg, Germany, pp.263-272, April, 2001.

[11] 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 Proc. Int'l Conf. on Management of Data, ACM SIGMOD, Madison, Wisconsin, pp.382-393, June, 2002.

[12] Oppenheim, A. V. and Schafer, R. W., Digital Signal Processing, Prentice-Hall, 1975.

[13] Park, S., Chu, W. W., Yoon, J., and Won, J., "Similarity Search of Time-Warped Subsequences via a Suffix Tree," Information Systems, Vol.28, No.7, pp.867-883, Oct., 2003.

[14] Rafiei, D., "On Similarity-Based Queries for Time Series Data," In Proc. the 15th Int'l Conf. on Data Engineering(ICDE), IEEE, Sydney, Australia, pp.410-417, Feb., 1999.

[15] Rafiei, D. and Mendelzon, A. O., "Querying Time Series Data Based on Similarity," IEEE Trans. on Knowledge and Data Engineering, Vol.12, No.5, pp.675-693, Sept./Oct., 2000.

[16] Wu, H., Salzberg, B., and Zhang, D., "Online Event-driven Subsequence Matching Over Financial Data Streams," In Proc. of Int'l Conf. on Management of Data, ACM SIGMOD, Paris, France, pp.23-34, June, 2004.

[17] Yi, B.-K., Jagadish, H. V., and Faloutsos, C., "Efficient Retrieval of Similar Time Sequences Under Time Warping," In Proc. the 14th Int'l Conf. on Data Engineering(ICDE), IEEE, Orlando, Florida, pp.201-208, Feb., 1998.

[부록 A]

[보조정리 3의 증명] 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$를 시퀀스 X라 표현하자. 즉, $S[k] = X[k-i+1]\,(i \le k \le j)$이라 하자. 그러면, 다음 과정에 의하여 증명이 완료된다.

$$D(\overline{S[i:j]}, \overline{Q}) \le \varepsilon \Leftrightarrow D(\overline{X}, \overline{Q}) \le \varepsilon$$

$$\Rightarrow D(\overline{X}[a-i+1:b-i+1], \overline{Q}[a-i+1:b-i+1]) \le \varepsilon$$

(보조정리 2에 의함)

$$\Leftrightarrow D(\overline{S[i:j][a:b]}, \overline{Q}[a-i+1:b-i+1]) \le \varepsilon$$

(X 의 정의에 의함)

$$\Leftrightarrow D(\overline{S_{[i:j]}[a:b]}, \overline{Q}[a-i+1:b-i+1]) \le \varepsilon$$

($\overline{S_{[i:j]}[a:b]}$의 정의에 의함) □

[부록 B]

[보조정리 4의 증명] 보조정리 4는 보조정리 1 및 3에 의해 다음 과정으로 증명이 완료된다.

$$D(\overline{S[i:j]}, \overline{Q}) \le \varepsilon$$

$$\Rightarrow D(\overline{S_{[i:j]}[a_0 : a_p - 1]}, \overline{Q}[a_0 - i + 1 : a_p - i]) \le \varepsilon$$

(보조정리 3에 의함)

$$\Rightarrow \bigvee_{k=1}^{p} D(\overline{S_{[i:j]}[a_{k-1} : a_k - 1]}, \overline{Q}[a_{k-1} - i + 1 : a_k - i]) \le \varepsilon/\sqrt{p}$$

(보조정리 1에 의함) □

문 양 세

e-mail : ysmoon@kangwon.ac.kr

1991. 2 한국과학기술원 과학기술대학 전산학과(학사)

1993. 2 한국과학기술원 전산학과(석사)

2001. 8 한국과학기술원 전자전산학과 전산학전공(박사)

1993. 2~1997. 2 현대전자산업(주) 통신사업본부 주임연구원

2001. 9~2002. 2 ㈜현대시스콤 호처리개발실 선임연구원

2002. 2~2005. 2 ㈜인프라밸리 기술연구소 기술위원(이사)

2005. 3~현재 한국과학기술원 첨단정보기술연구센터 연구원

2005. 3~현재 강원대학교 컴퓨터과학과 조교수

관심분야 : Data Mining, Knowledge Discovery, Stream Data, Storage System, Database Applications, Mobile/Wireless Communication Services & Systems

김 진 호

e-mail : jhkim@kangwon.ac.kr
1982. 2 경북대학교 전자공학과(학사)
1985. 2 한국과학기술원 전산학과(석사)
1990. 2 한국과학기술원 전산학과(박사)
1995. 8~1996.7 미국 미시간 대학교 객원 교수
2003. 2~2004. 2 미국 Drexel University 객원 교수
1999. 3~현재 한국과학기술원 첨단정보기술연구센터 연구원
1990. 8~현재 강원대학교 컴퓨터과학과 교수
관심분야 : Data warehouse, OLAP, Data Mining, Real-time/Embedded Database, Main-memory database, Data Modeling, Web Database Technology

노 웅 기

e-mail : woong@mozart.kaist.ac.kr
1991. 2 한국과학기술원 전산학과(학사)
1993. 2 한국과학기술원 전산학과(석사)
2001. 2 한국과학기술원 전산학과(박사)
2001. 2~2003. 9 (주)티맥스소프트 책임연구원
2003. 10~2005. 3 (주)티맥스데이타 수석연구원
2005. 4~현재 한국과학기술원 초빙교수
관심분야 : 데이터 마이닝/데이터 웨어하우징, 스트림 데이터 마이닝, 웹 마이닝, 정보 검색, 멀티미디어 데이터베이스, 멀티미디어 내용기반 검색, 공간 데이터베이스, 임베디드 데이터베이스, 데이터베이스 시스템 엔진