단일 색인을 사용한 임의 계수의 이동평균 변환 지원 시계열 서브시퀀스 매칭

(A Single Index Approach for Time-Series Subsequence Matching that Supports Moving Average Transform of Arbitrary Order)

문 양 세 † 김 진 호 †
(Yang-Sae Moon) (Jinho Kim)

요 약 본 논문에서는 단일 색인을 사용하는 임의 계수의 이동평균 변환 지원 서브시퀀스 매칭 방법을 제안한다. 단일 색인을 사용함으로써, 제안한 방법은 색인 저장 공간 및 색인 관리의 오버헤드를 크게 줄일 수 있다. 이동평균 변환은 시계열 데이타 내의 노이즈 영향을 감소시킴으로써, 시계열 데이타 전체의 경향을 파악하는데 매우 유용하다. 그런데, 기존 연구에서는 임의 계수를 지원하기 위해 여러 색인을 생성해야 하고, 이에 따라 색인 저장 공간의 오버헤드와 색인 관리의 오버헤드가 발생하는 문제점이 있다. 본 논문에서는 우선 이동평균 변환의 정의를 확장한 **다계수 이동평균 변환**(*poly-order moving average transform*) 개념을 제시한다. 다계수 이동평균 변환이란, 각 윈도우를 하나의 이동평균 계수에 대해서 이동평균 변환하는 것이 아니라, 여러 계수에 대해서 이동평균 변환하여 윈도우의 집합을 구성하는 변환으로서, 이동평균 변환의 정의를 여러 계수로 구성된 집합에 대해서 확장한 것이다. 다음으로, 이러한 다계수 이동평균 변환 개념을 사용한 서브시퀀스 매칭 방법의 이론적 근거인 정확성을 정리로서 제시하고 증명한다. 또한, 다계수 이동평균 변환을 기존 서브시퀀스 매칭 연구인 Faloutsos 등의 방법 및 DualMatch에 각각 적용하여, 두 가지 이동평균 변환 지원 서브시퀀스 매칭 방법을 제시한다. 실험 결과, 제안한 두 가지 서브시퀀스 매칭 방법은 모든 경우에 있어서 순차 스캔보다 성능을 크게 향상시킨 것으로 나타났다. 실제 주식 데이타에 대한 실험 결과, 제안한 방법은 순차 스캔에 비해서 평균 22.4배~33.8배까지 성능을 향상시킨 것으로 나타났다. 또한, 각 계수에 대해 모두 색인을 생성하는 경우와 비교할 때, 성능 저하는 매우 적은 반면 필요한 색인 공간은 크게 줄인 것으로 나타났다(일곱 개의 계수를 사용한 경우, 성능 저하는 평균 9%~42%에 불과한 반면 색인 공간은 약 1/7.0로 크게 줄인다). 이와 같이 성능 측면과 색인 공간 및 관리 측면에서의 우수성에 덧붙여, 제안한 방법은 이동평균 변환 이외의 다른 변환을 지원하는 서브시퀀스 매칭으로 일반화 될 수 있는 장점이 있다. 따라서, 제안한 방법은 이동평균 변환을 포함하는 많은 다른 종류의 변환을 지원하는 서브시퀀스 매칭에 폭넓게 적용되는 우수한 연구결과라 사료된다.

키워드 : 데이타 마이닝, 시계열 데이타, 서브시퀀스 매칭, 이동평균 변환

Abstract We propose a single index approach for subsequence matching that supports moving average transform of arbitrary order in time-series databases. Using the single index approach, we can reduce both storage space overhead and index maintenance overhead. Moving average transform is known to reduce the effect of noise and has been used in many areas such as econometrics since it is useful in finding overall trends. However, the previous research results have a problem of occurring index overhead both in storage space and in update maintenance since the methods build several indexes to support arbitrary orders. In this paper, we first propose the concept of *poly-order moving average transform*, which uses a set of order values rather than one order value, by extending the original definition of moving average transform. That is, the poly-order transform makes a set of transformed windows from each original window since it transforms each window not for just one

· 본 연구는 첨단정보기술연구센터를 통하여 과학기술부/한국과학재단의 지원을 받았음
† 정 회 원 : 강원대학교 컴퓨터과학과 교수
ysmoon@kangwon.ac.kr
jhkim@kangwon.ac.kr
논문접수 : 2005년 7월 15일
심사완료 : 2005년 10월 19일

order value but for a set of order values. We then present theorems to formally prove the correctness of the poly-order transform based subsequence matching methods. Moreover, we propose two different subsequence matching methods supporting moving average transform of arbitrary order by applying the poly-order transform to the previous subsequence matching methods. Experimental results show that, for all the cases, the proposed methods improve performance significantly over the sequential scan. For real stock data, the proposed methods improve average performance by 22.4~33.8 times over the sequential scan. And, when comparing with the cases of building each index for all moving average orders, the proposed methods reduce the storage space required for indexes significantly by sacrificing only a little performance degradation(when we use 7 orders, the methods reduce the space by up to 1/7.0 while the performance degradation is only 9%~42% on the average). In addition to the superiority in performance, index space, and index maintenance, the proposed methods have an advantage of being generalized to many sorts of other transforms including moving average transform. Therefore, we believe that our work can be widely and practically used in many sort of transform based subsequence matching methods.

Key words : Data Mining, Time Series Data, Subsequence Matching, Moving Average Transform

1. 서 론

시계열 데이타(time-series data)란 각 시간별로 측정한 실수 값의 시퀀스로, 그 예로는 주식 데이타, 환율 데이타, 날씨 변동 데이타 등이 있다[1-3]. 시계열 데이타베이스에 저장된 시계열 데이타를 **데이타 시퀀스**(*data sequence*)라 부르며, 사용자에 의해 주어진 시퀀스를 **질의 시퀀스**(*query sequence*)라 부른다. 그리고, 주어진 질의 시퀀스와 유사한 데이타 시퀀스를 검색하는 방법을 **유사 시퀀스 매칭**(*similar sequence matching*)이라 한다[1,2]. 일반적으로, 유사 시퀀스 매칭에서는 길이 n인 두 시퀀스 X(={X[1], X[2], ···, X[n]})와 Y(={Y[1], Y[2], ···, Y[n]})의 거리가 사용자가 제시한 **허용치**(*tolerance*)인 ε 이하이면, 두 시퀀스 X와 Y는 **유사**(*similar*)하다고 정의한다[1,2,4]. 그리고, 본 논문에서는 거리 함수 D(X,Y)로 유클리디안 거리 함수 ($=\sqrt{\sum_{i=1}^{n}(X[i]-Y[i])^2}$)를 사용하며[1,2,5,6], $D(X,Y)$가 ε 이하이면 X와 Y는 **ε-매치**(*ε-match*)한다고 정의한다[4].

유사 시퀀스 매칭은 크게 전체 매칭(whole matching)과 서브시퀀스 매칭(subsequence matching)의 두 가지로 구분한다[2]. 전체 매칭은 질의 시퀀스와 유사한 데이타 시퀀스를 찾는 문제로서, 질의 시퀀스와 데이타 시퀀스의 길이가 동일한 특징을 갖는다[1]. 반면에, 서브시퀀스 매칭은 데이타 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[2-4,6-8]. 그리고, 유클리디안 거리 함수가 갖는 문제점을 보완하기 위하여, 이동평균(moving average)[7,9], 쉬프팅 및 스케일링(shifting & scaling)[10-12], 정규화(normalization)[6,9], 타임 워핑(time warping)[13-15] 등의 다양한 변환 기법이 사용되었다. 본 논문에서는 이들 변환 중에서 이동평균 변환을 지원하는 서브시퀀스 매칭 문제를 다룬다.

이동평균 변환, 정확히는 k-이동평균 변환은 주어진 시퀀스에서 연속된 k개 엔트리의 평균 값을 각 엔트리로 하는 새로운 시퀀스를 구성하는 변환이다. 여기에서, k 값을 **이동평균 계수**(*moving average order*) 또는 간략히 **계수**(*order*)라 한다. 이러한 이동평균 변환은 시계열 데이타 내의 노이즈 영향을 감소시킴으로써 시계열 데이타 전체의 경향을 파악하는데 매우 유용한 것으로 알려져 있다[16]. 그리고, 응용 분야와 시계열 데이타의 특성에 따라 노이즈의 영향을 줄이고자 하는 정도와 경향을 파악하고자 하는 주기가 달라지므로, 조건에 따라 여러 값의 이동평균 계수가 사용될 수 있다[17]. 예를 들어, 주가 데이타의 경향을 파악하는데 사용되는 이동평균 계수 값은 주로 6, 25, 75, 150이며, 작은 값일수록 짧은 주기의 경향 파악에, 큰 값일수록 긴 주기의 경향 파악에 많이 사용된다[7]. 이와 같이, 응용이나 데이타에 따라 사용되는 이동평균 계수가 달라지므로, 유사 시퀀스 매칭에서는 임의의 이동평균 계수를 지원할 수 있어야 한다[7,9].

본 논문에서는 이동평균 변환을 서브시퀀스 매칭에 적용하는 유사 시퀀스 모델[7]을 다룬다. 즉, 질의 시퀀스와 데이타 서브시퀀스의 거리를 비교하는 것이 아니라, 이들을 주어진 계수 k에 의해 k-이동평균 변환한 이후의 두 시퀀스를 비교하여, 변환된 두 시퀀스가 ε-매치하는지의 여부를 판단하는 유사 시퀀스 모델을 다룬다. 이와 같이 임의 계수의 이동평균 변환을 지원하는 서브시퀀스 매칭 방법을 본 논문에서는 간략히 **이동평**

균 변환 서브시퀀스 매칭이라 부른다.

본 논문에서는 단일 색인을 사용하는 이동평균 변환 서브시퀀스 매칭 방법을 제안한다. 기존의 서브시퀀스 매칭 알고리즘[2,8]은 이동평균 변환 서브시퀀스 매칭에 그대로 적용될 수 없다[7]. 그리고, 이동평균 변환 서브시퀀스 매칭을 처음 제안한 Loh 등[7]의 방법은 기존 다차원 색인의 구조와 알고리즘을 변경해야 하는 문제점과 여러 색인에 따른 저장 공간 및 색인 관리의 오버헤드가 발생하는 문제점이 있다. 반면에, 본 논문에서는 하나의 색인을 사용하면서도, 임의 계수에 대한 이동평균 변환 서브시퀀스 매칭을 효율적으로 수행하는 방법을 제안한다. 이를 위하여, 우선 윈도우를 여러 계수로 구성된 계수 집합에 대해서 이동평균 변환하는 **다계수 이동평균 변환**(*poly-order moving average transform*) 개념을 제시한다. 다계수 이동평균 변환이란, 각 윈도우를 특정 계수에 대해서 이동평균 변환하는 것이 아니라, 여러 계수에 대해서 이동평균 변환하여 윈도우의 집합을 구성하는 변환을 의미한다. 이러한 다계수 이동평균 변환을 사용하는 이유는 이동평균 변환에 사용되는 계수 값이 다르더라도, 각 계수에 따라 변환된 윈도우들은 유사한 엔트리 값을 가지는 이동평균 변환의 정의에 기반한다. 본 논문에서는 다계수 이동평균 변환을 사용하여 다차원 색인을 구성하면, 하나의 색인을 사용해서도 이동평균 변환 서브시퀀스 매칭을 수행할 수 있음을 보인다. 또한, 다계수 이동평균 변환을 사용하면, 기존 서브시퀀스 매칭에서 사용하였던 이론을 그대로 이동평균 변환 서브시퀀스 매칭에 적용할 수 있음을 정리로서 제시하고 증명한다.

다음으로, 다계수 이동평균 변환의 개념을 서브시퀀스 매칭의 기존 연구인 Faloutsos 등의 연구[2](간략히, ***FRM***이라 한다) 및 DualMatch[8]에 적용하는 새로운 이동평균 변환 서브시퀀스 매칭 방법을 제안한다. 우선, FRM에서 취한 데이타 시퀀스를 슬라이딩 윈도우로 나누는 방법에 다계수 이동평균 변환 개념을 적용하여 다차원 색인을 구성하는 방법을 제안하고, 구성된 다차원 색인을 사용하여 이동평균 변환 서브시퀀스 매칭을 수행하는 방법을 제안한다. 다음으로, 제안한 다계수 이동평균 변환 개념을 사용하면, 데이타 시퀀스를 디스조인트 윈도우로 나누는 방법을 사용하는 DualMatch도 이동평균 서브시퀀스 매칭 방법으로 확장될 수 있음을 보인다. 그리고, 이들 FRM 및 DualMatch에 다계수 이동평균 변환을 적용한 색인 구성 알고리즘과 서브시퀀스 매칭 알고리즘을 각각 제시한다. 성능 평가 결과, 제안한 두 가지 서브시퀀스 매칭 방법은 선택률의 범위 및 질의 시퀀스의 길이에 관계없이 모든 경우에 있어서 순차 스캔보다 성능을 크게 향상시킨 것으로 나타났다. 실제 주식 데이타에 대한 실험 결과, 제안한 방법은 순차 스캔에 비해 성능을 평균 22.4배에서 33.8배까지 크게 향상시킨 것으로 나타났다. 또한, 이동평균 변환의 각 계수에 대해 모두 색인을 생성한 경우에 비해서 성능 저하는 매우 적은 반면에, 필요한 색인 공간은 크게 줄인 것으로 나타났다(일곱 개의 계수를 사용한 경우, 성능 저하는 평균 9%~42%에 불과한 반면 색인 공간은 약 1/7.0로 크게 줄인다).

본 논문의 구성은 다음과 같다. 제2장은 관련 연구로서, 기존의 유클리디안 서브시퀀스 매칭 방법 및 이동평균 변환 서브시퀀스 매칭 방법을 설명한다. 제3장에서는 제안하는 이동평균 변환 서브시퀀스 매칭을 개념, 정확성, 알고리즘의 순으로 설명한다. 제4장에서는 실험을 통해 제안한 방법의 우수성을 보이고, 마지막으로 제5장에서 결론을 맺는다.

2. 관련 연구

본 장에서는 유클리디안 거리 기반의 서브시퀀스 매칭과 이동평균 변환을 지원하는 유사 시퀀스 매칭에 관한 기존 연구를 설명한다. 제1장에서 언급한 바와 같이, 이동평균 변환을 제외한 다른 유사 시퀀스 모델에 대해서는 참고문헌[6,9-15]의 연구를 참조한다. 그리고, 이동평균 변환을 포함하여, 본 논문에서 사용하는 주요 표기와 이에 대한 정의 및 의미는 표 1과 같다. 표 1의 표기법에 있어서, 혼란이 없는 한 $S^{(k)}[i:j]$를 "서브시퀀스 $S[i:j]$를 k-이동평균변환한 서브시퀀스"라 부르고, $s_i^{(k)}$를 "윈도우 s_i를 k-이동평균변환한 윈도우"라 부른다.

기존의 서브시퀀스 매칭 방법들은 Agrawal 등[1]의 전체 매칭을 발전시켜 문제를 해결하였다. 그러므로, 우선 이러한 전체 매칭을 색인 구성 알고리즘과 유사 시퀀스 매칭 알고리즘으로 구분하여 설명한다. 색인 구성 알고리즘에서는 길이 n인 데이타 시퀀스에서 $f(\le n)$개의 특성(feature)을 추출하여 f-차원 공간의 점으로 **저차원 변환**(*lower-dimensional transformation*)[1,2]한 후, 이를 f-차원의 R*-트리[18]에 저장한다. 이렇게 특성을 추출하는 이유는 다차원 색인의 고차원 문제(high dimensionality problem)[19]로 인하여, 고차원인 시퀀스를 다차원 색인인 R*-트리에 직접 저장하기 어렵기 때문이다. 그리고, 이와 같이 저차원 변환을 위해 사용하는 함수를 **특성 추출 함수**(*feature extraction function*)라 한다[2,4,8]. 다음으로, 유사 시퀀스 매칭 알고리즘에서는 질의 시퀀스를 데이타 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 점

표 1 주요 표기법

기호	정의/의미
$Len(S)$	시퀀스 S의 길이
$S[i]$	시퀀스 S의 i번째 엔트리
$S[i:j]$	시퀀스 S의 i번째에서 j번째 엔트리까지로 구성된 서브시퀀스
$S^{(k)}$	시퀀스 S를 k-이동평균 변환한 시퀀스$\left(S^{(k)}[i]=\frac{1}{k}\sum_{j=i}^{i+k-1}S[j]\right)$
$S^{(k)}[i]$	k-이동평균 변환된 시퀀스 $S^{(k)}$의 i번째 엔트리
$S^{(k)}[i:j]$	k-이동평균 변환된 시퀀스 $S^{(k)}$의 i에서 j번째까지 엔트리로 구성된 서브시퀀스
s_i	시퀀스 S의 i번째 디스조인트 윈도우$(=S[(i-1)\cdot\omega+1:i\cdot\omega], i\geq 1)$
$s_i^{(k)}$	k-이동평균 변환된 시퀀스 $S^{(k)}$의 i번째 디스조인트 윈도우 $(=S^{(k)}[(i-1)\cdot\omega+1:i\cdot\omega], i\geq 1)$

들을 찾아 **후보**(*candidate*, 질의 시퀀스와 ε-매치할 가능성이 높은 데이타 시퀀스) 집합을 구한다. 이렇게 후보집합을 구하면 **착오기각**(*false dismissal*, 유사 시퀀스이나 착오로 인해 기각되는 데이타 시퀀스)은 발생하지 않지만, 시퀀스 길이 n대신 f개의 특성만을 사용함으로 인하여 **착오해답**(*false alarm*, 후보이나 실제로는 질의 시퀀스와 ε-매치하지 않는 데이타 시퀀스)이 발생할 수 있다. 따라서, R*-트리에 대한 검색 결과로 얻은 각 후보 시퀀스들에 대해서는 데이타베이스에 저장된 실제 데이타 시퀀스를 액세스하고 질의 시퀀스와의 거리를 조사하여 착오해답을 제거하는데, 이 과정을 **후처리 과정**(*post-processing step*)이라 한다[1].

Faloutsos 등[2]은 전체 매칭을 일반화하여 서브시퀀스 매칭을 처음 소개하고, 이의 해결책(FRM)을 제시하였다. FRM에서는 데이타 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 데이타 시퀀스를 나눈 슬라이딩 윈도우로 f-차원의 점으로 변환하여 다차원 색인인 R*-트리에 저장한다. 그런데, 데이타 시퀀스를 슬라이딩 윈도우로 나누기 때문에 너무 많은 점이 생성되는 문제점이 있다[2,6]. 이를 해결하기 위하여, FRM에서는 여러 개의 점을 포함하는 MBR(minimum bounding rectangle)을 구성하고, 이 MBR만을 다차원 색인인 R*-트리에 저장하는 방법을 사용한다. 다음으로, FRM은 다음 보조정리 1에 기반하여 서브시퀀스 매칭을 수행한다.

보조정리 1[2]. 동일한 길이의 시퀀스 S와 Q를 각각 p개의 디스조인트 윈도우 s_i와 $q_i(1\leq i\leq p, p=\lfloor Len(Q)/\omega \rfloor)$로 나누었을 때, 두 시퀀스 S와 Q가 ε-매치한다면, 적어도 하나 이상의 (s_i, q_i) 쌍이 $\varepsilon/\sqrt{p}$-매치한다. 즉, 다음 식 (1)이 성립한다.

$$D(S,Q)\leq\varepsilon \Rightarrow \bigvee_{i=1}^{p} D(s_i,q_i)\leq\varepsilon/\sqrt{p} \quad (1)$$

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

DualMatch[8]와 GeneralMatch[4]는 윈도우 구성법을 달리하여 FRM의 성능을 개선한 서브시퀀스 매칭 방법들이다. 우선, DualMatch에서는 윈도우 구성의 이원성(duality) 개념을 제시하고, 이원성에 기반하여 데이타 시퀀스를 디스조인트 윈도우로 나누고 질의 시퀀스를 슬라이딩 윈도우로 나누는 FRM의 이원적 접근법을 제안하였다. 다음으로, GeneralMatch에서는 FRM과 DualMatch에서 사용한 슬라이딩 윈도우와 디스조인트 윈도우를 일반화한 J-슬라이딩 윈도우와 J-디스조인트 윈도우 개념을 제시하고, 이들 일반화된 윈도우를 사용한 서브시퀀스 매칭 방법을 제안하였다. 이들 서브시퀀스 매칭 방법의 색인 구성 및 서브시퀀스 매칭 알고리즘은 윈도우 구성을 달리하는 것을 제외하고는 FRM의 알고리즘과 유사하다.

Rafiei 등[9]은 임의 계수의 이동평균 변환을 지원하는 전체 매칭 알고리즘을 제시하였다. 그러나, Rafiei 등은 임의 계수의 이동평균 변환을 지원하기 위하여, 전통적인 이동평균 변환 정의가 아닌 다른 정의를 사용하였다. 즉, 길이 n인 시퀀스를 이동평균 변환할 때, k-이동평균 값을 구할 수 없는 마지막 $(k-1)$개의 데이타 값들에 대해서 부족한 $(k-1)$개의 값들을 시퀀스의 맨 앞에서 가져와 순환적(circular)으로 k-이동평균 값들을 구하였다. 이와 같이, 전통적인 이동평균의 정의를 사용하

지 않음으로써, Rafiei 등의 전체 매칭 연구는 서브시퀀스 매칭에 그대로 적용될 수 없다[7]. 즉, Rafiei 등의 전체 매칭 연구는 기존의 서브시퀀스 매칭에서 사용한 긴 시퀀스를 여러 개의 윈도우로 나누어 저장 및 검색하는 방법[2,4,8]에 적용할 수 없게 된다. 왜냐하면, Rafiei 등이 사용한 변형된 이동평균 정의를 사용하면, 대상 시퀀스 전체를 변환하는가 또는 윈도우로 변환하는가에 따라서 변환 결과가 달라지기 때문이다[7].

Loh 등[7]은 임의 계수의 이동평균 변환을 지원하는 서브시퀀스 매칭을 처음으로 제안하였다. Loh 등의 연구에서는 데이타 시퀀스를 m-이동평균 변환하고, 이를 윈도우로 나누어 저차원 변환한 후 다차원 색인에 저장하여 m-인덱스를 구성하였다. 그리고, 주어진 계수 k에 대한 k-이동평균 변환 서브시퀀스 매칭을 수행하기 위하여, 이미 구성된 m-인덱스를 사용하였다. 그런데, 이 방법은 임의계수 이동평균 변환을 지원하기 위하여 기존 다차원 색인의 구조와 알고리즘을 변경해야 하므로, 실용적으로 사용하기가 어려운 문제점이 있다. 또한, 원하는 계수 k 값이 인덱스가 구성된 m과 다른 경우, 색인의 검색 범위가 넓어지는 문제점이 있다. 이러한 검색 범위가 넓어지는 문제점을 해결하기 위하여, Loh 등은 여러 개의 계수에 대해서 m-인덱스를 구성하는 인덱스 보간법(index interpolation)[6]을 제안하였다. 그러나, 이러한 인덱스 보간법은 여러 색인의 사용에 따른 색인 저장 공간의 오버헤드와 데이타의 추가 및 삭제 시 색인 관리 오버헤드가 발생하는 문제점을 야기한다.

3. 단일 색인을 사용한 이동평균 변환 서브시퀀스 매칭

본 장에서는 단일 색인을 사용하는 이동평균 변환 서브시퀀스 매칭을 제안한다. 제3.1절에서는 다계수 이동평균 변환의 개념과 이를 사용하는 서브시퀀스 매칭의 정확성을 설명한다. 그리고, 제3.2절과 제3.3절에서는 다계수 이동평균 변환을 기존 서브시퀀스 매칭인 FRM에 적용한 ***FRM-MAT***(*FRM with Moving Average Transform*)과 DualMatch에 적용한 ***DM-MAT***(*Dual-Match with Moving Average Transform*)을 각각 제안한다.

3.1 개념

본 연구의 동기는 FRM과 DualMatch와 같은 기존 서브시퀀스 매칭에서 사용했던 보조정리 1의 수식 (2)를 이동평균 변환 서브시퀀스 매칭에 그대로 활용하자는데 있다. 보조정리 1을 사용할 경우, 색인 검색 범위가 $\varepsilon/\sqrt{p}$ 으로 줄어들어 효과적인 서브시퀀스 매칭을 수행할 수 있다. 따라서, 본 논문에서는 보조정리 1을 k-이동평균 변환에 적용한 다음의 보조정리 2를 제시한다.

보조정리 2. 동일한 길이의 시퀀스 S와 Q를 이동평균 계수 k로 k-이동평균 변환한 두 시퀀스 $S^{(k)}$와 $Q^{(k)}$를 각각 p개의 디스조인트 윈도우 $s_i^{(k)}$와 $q_i^{(k)}$ ($1 \le i \le p, p = \lfloor Len(Q^{(k)})/\omega \rfloor$)로 나누었을 때, 두 시퀀스 $S^{(k)}$와 $Q^{(k)}$가 ε-매치한다면, 적어도 하나 이상의 ($s_i^{(k)}, q_i^{(k)}$) 쌍이 $\varepsilon/\sqrt{p}$ -매치한다. 즉, 다음 식 (2)가 성립한다.

$$D(S^{(k)}, Q^{(k)}) \le \varepsilon \Rightarrow \bigvee_{i=1}^{p} D(s_i^{(k)}, q_i^{(k)}) \le \varepsilon/\sqrt{p} \quad (2)$$

증명. 시퀀스 $S^{(k)}$와 $Q^{(k)}$를 각각 X와 Y라 표현하자. 그러면, 보조정리 1에 의해서 다음 식 (3)이 성립한다.

$$D(X, Y) \le \varepsilon \Rightarrow \bigvee_{i=1}^{p} D(x_i, y_i) \le \varepsilon/\sqrt{p} \quad (3)$$

식 (3)에서, x_i와 y_i는 표 1의 표기법에 따라서 각각 $s_i^{(k)}$와 $q_i^{(k)}$로 표현할 수 있다. 따라서, 식 (3)은 식 (2)로 표현될 수 있고, 결국 보조정리 2가 성립한다. □

그런데, 보조정리 2를 사용하여 이동평균 변환 서브시퀀스 매칭을 수행하기 위해서는 가능한, 즉 주어질 수 있는 모든 k에 대해서 다차원 색인을 구성해야 한다. 즉, 각각의 k에 대해서, 데이타 시퀀스들이 k-이동평균 변환된 시퀀스들을 대상으로 별개의 다차원 색인을 구성해야 한다. 이 방법은 기존 서브시퀀스 매칭 방법[2,4,8]을 그대로 활용할 수 있어 구현이 용이하나, 모든 k에 대해서 별도의 다차원 색인을 구성해야 하므로, 색인 공간의 오버헤드와 색인 관리의 오버헤드가 발생하는 문제점이 있다[7]. 이와 같은 문제점을 해결하기 위하여, 본 논문에서는 하나의 색인을 사용하면서도 임의 계수의 이동평균 변환을 지원하는 서브시퀀스 매칭 방법을 제안한다.

임의 계수 이동평균 변환을 FRM과 DualMatch와 같은 기존 서브시퀀스 매칭 방법에 적용하기 위하여, 본 논문에서는 다음 정의 1과 같이 k-이동평균 변환의 개념을 확장한다.

정의 1. 시퀀스 S에 포함된 윈도우 $S[a:b]$를 이동평균 계수의 집합 $K = \{k_1, k_2, ..., k_m\}$으로 **다계수 이동평균 변환**(poly-order moving average transform)한 집합 $S^{(K)}[a:b]$는 다음 식 (4)와 같이 정의한다.

$$S^{(K)}[a:b] = \{S^{(k_i)}[a:b] \mid 1 \le i \le m\} \quad (4)$$

정의 1을 다시 설명하면, 윈도우 $S[a:b]$를 집합 K로 다계수 이동평균 변환한 집합 $S^{(\mathrm{K})}[a:b]$는, K에 포함된 각각의 계수 k_i로 $S[a:b]$를 k_i-이동평균 변환한 윈도우인 $S^{(k_i)}[a:b]$를 원소로 포함하는 집합이다. 그리고, 여러 윈도우들을 포함하는 영역을 표현하기 위하여 윈도우 집합에 대한 MBR을 다음과 같이 정의한다.

정의 2. 크기 ω인 윈도우의 집합 $\mathrm{W}=\{W_1, W_2, ..., W_m\}$에 대해, W의 원소인 윈도우 W_i들을 모두 포함하는 ω-차원의 MBR을 집합 W의 **MBR**이라 정의하고, 이를 $\boldsymbol{MBR}(\mathbf{W})$로 표기한다.

정의 1 및 정의 2를 사용하여, 시퀀스 S에 포함된 i번째 디스조인트 윈도우인 s_i를 집합 K로 다계수 이동평균 변환한 집합을 $s_i^{(\mathrm{K})}$라 표현한다. 그리고, 이 집합 $s_i^{(\mathrm{K})}$의 모든 윈도우들을 포함하는 MBR은 $MBR(s_i^{(\mathrm{K})})$이라 표현한다.

다계수 이동평균 변환을 사용하면, 정확하게, 즉 착오기각이 발생하지 않게 이동평균 변환 서브시퀀스 매칭을 수행할 수 있다. 이를 설명하기 위하여 우선 보조정리 3을 제시한다. 보조정리 3은 k-이동평균 변환과 k를 포함하는 집합 K에 대한 다계수 이동평균 변환과의 관계를 나타낸다.

보조정리 3. 이동평균 계수 k가 집합 K의 원소라 할 때, 즉 $k \in \mathbf{K}$라 할 때, 시퀀스 S와 Q를 k-이동평균 변환한 $S^{(k)}$와 $Q^{(k)}$의 i번째 디스조인트 윈도우인 $s_i^{(k)}$와 $q_i^{(k)}$가 ε-매치한다면, $q_i^{(k)}$는 $MBR(s_i^{(\mathrm{K})})$와 ε-매치한다. 즉, 다음 식 (5)가 성립한다.

$$D(q_i^{(k)}, s_i^{(k)}) \le \varepsilon \Rightarrow D(q_i^{(k)}, MBR(s_i^{(\mathbf{K})})) \le \varepsilon \quad (5)$$

증명. 윈도우 집합 W에 대한 $MBR(\mathbf{W})$의 정의에 따라 증명할 수 있다. 우선 윈도우 크기를 ω라 하자. 그러면, $MBR(\mathbf{W})$의 정의에 따라서, ω-차원의 점에 해당하는 $s_i^{(k)}$는 ω-차원의 MBR인 $MBR(s_i^{(\mathrm{K})})$에 포함된다. 따라서, $q_i^{(k)}$와 $MBR(s_i^{(\mathrm{K})})$의 거리는 $q_i^{(k)}$와 $s_i^{(k)}$의 거리 이하가 된다. 즉, $D(q_i^{(k)}, MBR(s_i^{(\mathbf{K})})) \le D(q_i^{(k)}, s_i^{(k)})$이 성립하고, 결국 식 (5)가 성립한다. □

다음으로, 보조정리 2와 3에 따라서, 이동평균 변환 서브시퀀스 매칭의 이론적 근거가 되는 다음 정리 1이 성립한다.

정리 1. 이동평균 계수 k가 집합 $\mathbf{K}$의 원소라 할 때, 즉 $k \in \mathbf{K}$라 할 때, 질의 시퀀스 Q를 k-이동평균 변환한 $Q^{(k)}$가 데이타 시퀀스 S의 서브시퀀스 $S[a:b]$ ($Len(S[a:b]) = Len(Q^{(k)})$)를 k-이동평균 변환한 $S^{(k)}[a:b]$와 ε-매치한다면, 적어도 하나 이상의 $Q^{(k)}$에 포함된 디스조인트 윈도우 $q_i^{(k)}$는 $S[a:b]$에 포함된 윈도우 $S[a+(i-1)\cdot\omega : a+i\cdot\omega-1]$를 $\mathbf{K}$에 대해 다계수 이동평균 변환한 윈도우들의 MBR인 $MBR(S^{(\mathbf{K})}[a+(i-1)\cdot\omega : a+i\cdot\omega-1])$과 $\varepsilon/\sqrt{p}$-매치한다. 즉, 다음 식 (6)이 성립한다. 여기에서, $p = \lfloor Len(Q^{(k)})/\omega \rfloor$이다.

$$D(Q^{(k)}, S^{(k)}[a:b]) \le \varepsilon \Rightarrow \bigvee_{i=1}^{p} D(q_i^{(k)}, MBR(S^{(\mathrm{K})}[a+(i-1)\cdot\omega : a+i\cdot\omega-1])) \le \varepsilon/\sqrt{p} \quad (6)$$

증명. 이동평균 변환된 서브시퀀스 $S^{(k)}[a:b]$를 시퀀스 $X^{(k)}$라 표현하자. 즉, $X^{(k)}[j] = S^{(k)}[a+j-1]$이고, $1 \le j \le b-a+1$라 하자. 그러면, 다음 과정에 의하여 증명이 완료된다.

$D(Q^{(k)}, S^{(k)}[a:b]) \le \varepsilon$

$\Leftrightarrow D(Q^{(k)}, X^{(k)}) \le \varepsilon$ ($X^{(k)}$의 정의에 의함)

$\Rightarrow \bigvee_{i=1}^{p} D(q_i^{(k)}, x_i^{(k)}) \le \varepsilon/\sqrt{p}$ (보조정리 2에 의함)

$\Leftrightarrow \bigvee_{i=1}^{p} D(q_i^{(k)}, MBR(x_i^{(\mathbf{K})})) \le \varepsilon/\sqrt{p}$ (보조정리 3에 의함)

$\Leftrightarrow \bigvee_{i=1}^{p} D(q_i^{(k)}, MBR(S^{(\mathrm{K})}[a+(i-1)\cdot\omega : a+i\cdot\omega-1])) \le \varepsilon/\sqrt{p}$ ($X^{(k)}$의 정의에 의함) □

결국, 정리 1을 사용하게 되면, 서브시퀀스 매칭 과정에 있어서 $q_i^{(k)}$와 $\varepsilon/\sqrt{p}$-매치하는 $MBR(S^{(\mathbf{K})}[a+(i-1)\cdot\omega : a+i\cdot\omega-1])$에 대해서, 즉 식 (6)의 필요조건이 만족하는 경우에 대해서, 서브시퀀스 $S^{(k)}[a:b]$를 $Q^{(k)}$와 ε-매치하는 후보 서브시퀀스로 삼으면, 착오기각이 발생하지 않고 k-이동평균 변환 서브시퀀스 매칭을 정확하게 수행할 수 있다.

정리 1을 기존 서브시퀀스 매칭에 적용하는 방법을 설명하면 다음과 같다. 기존 서브시퀀스 매칭인 FRM이나 DualMatch에서는 데이타 시퀀스를 윈도우로 나누고,

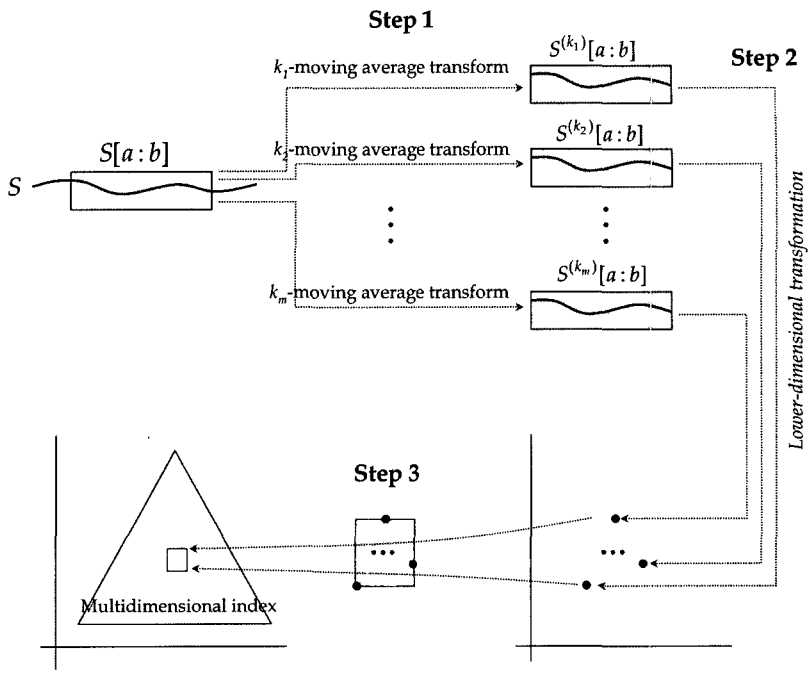


그림 1 다계수 이동평균 변환을 적용한 각 윈도우의 다차원 색인 저장 과정

이들 윈도우를 저차원 변환하여 다차원 색인에 저장하는 방법을 사용하였다. 반면에, 임의 계수의 이동평균 변환을 지원하기 위해서는, 각각의 윈도우를 가능한 모든 계수들의 집합으로 다계수 이동평균 변환하여 윈도우들의 집합을 구성하여야 한다. 즉, 하나의 윈도우가 각각의 계수에 따라서 여러 개의 윈도우들로 구성되는 집합으로 변환된다. 따라서, 제안하는 방법에서는 이들 윈도우들을 포함하는 MBR을 저차원 변환하여[1] 다차원 색인에 저장하는 방법을 사용한다. 그림 1은 다계수 이동평균 변환 및 저차원 변환을 통하여 하나의 윈도우에 대해서 색인에 저장할 MBR이 구성되는 과정을 나타낸다. 그림에서 보면, 단계 1에서는 데이타 시퀀스 S를 나눈 각 윈도우 $S[a:b]$를 각 계수 k_i를 사용하여 이동평균 변환한 $S^{(k_i)}[a:b]$들을 생성한다. 다음으로, 단계 2에서는 변환된 윈도우들을 저차원 변환하여 저차원 공간의 점으로 변환한다. 마지막으로, 단계 3에서는 이들 점을 포함하는 MBR을 구성하고, 이 MBR을 다차원 색인에 저장하여 서브시퀀스 매칭에 사용하게 된다. 이와 같은 다계수 이동평균 변환을 사용하는 방법이 가능한 이유는 윈도우 $S[a:b]$를 $k_i(\in \mathbf{K})$-이동평균 변환한 $S^{(k_i)}[a:b]$와 $k_j(\in \mathbf{K})$-이동평균 변환한 $S^{(k_j)}[a:b]$가 유사한 값을 가질 것이라는 관찰에 근거한다. 즉, k_i-이동평균 변환과 k_j-이동평균 변환은 변환에 사용하는 엔트리 개수가 k_i개와 k_j개로 다를 뿐, 상당수의 동일한 엔트리가 공통적으로 사용되기 때문에, 두 계수로 변환된 윈도우의 엔트리 값은 유사하다는 관찰에 근거한다.

결국, 정리 1을 사용하면 기존의 서브시퀀스 매칭 방법을 확장하여 이동평균 변환 서브시퀀스 매칭을 구현할 수 있다. 먼저, 데이타 시퀀스를 슬라이딩 윈도우로 나누는 윈도우 구성법을 사용하는 FRM의 경우는 각각의 슬라이딩 윈도우를 다계수 이동평균 변환과 저차원 변환으로 구성한 MBR을 다차원 색인에 저장하는 방법을 사용한다. 다음으로, 데이타 시퀀스를 디스조인트 윈도우로 나누는 DualMatch의 경우는 각각의 디스조인트 윈도우를 다계수 이동평균 변환과 저차원 변환으로 구성한 MBR을 다차원 색인에 저장하는 방법을 사용한다. 다음의 제3.2절과 제3.3절에서는 FRM과 DualMatch를 다계수 이동평균 변환 개념을 사용하여 확장한 이동평균 변환 서브시퀀스 매칭을 수행하는 색인 구성 알고리

1) MBR에 대한 저차원 변환은 정의되지 않았으므로, 실제로는 각 윈도우를 저차원 변환한 후, 저차원 변환된 점들을 포함하는 MBR을 구성한다.

Procedure *FRM-MAT-BuildIndex*(Data Sequence S, Window size ω, Set of orders K)

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

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

(3) Make a set of windows $S^{(K)}[a:b]$ by using the *poly-order moving average transform* on K;

(4) Construct an f-dimensional MBR f-*D MBR* by using lower-dimensional transformations on $S^{(K)}[a:b]$;

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

(6) **endfor**

그림 2 FRM-MAT의 다차원 색인 구성 알고리즘

즘 및 서브시퀀스 매칭 알고리즘을 각각 설명한다.[2)]

3.2 FRM-MAT: FRM with Moving Average Transform

본 절에서는 FRM의 서브시퀀스 매칭 방법에 다계수 이동평균 변환을 적용하여 임의 계수 이동평균 변환을 지원하는 서브시퀀스 매칭 방법을 설명한다. FRM에서는 데이타 시퀀스를 슬라이딩 윈도우로 나누고, 질의 시퀀스를 디스조인트 윈도우로 나누는 윈도우 구성법을 사용한다. FRM-MAT는 이러한 윈도우 구성법에 다계수 이동평균 변환을 적용한 이동평균 변환 서브시퀀스 매칭 방법이다.

그림 2는 FRM-MAT의 색인 구성 알고리즘을 나타낸다. 우선, 단계 (1)에서는 주어진 데이타 시퀀스를 크기 ω의 슬라이딩 윈도우로 나눈다. 다음으로, 단계 (2)~(6)에서는 각 윈도우를 다차원 색인의 하나의 MBR로 저장하는 작업을 수행한다. 먼저, 단계 (3)에서는 각 슬라이딩 윈도우에 대해서 다계수 이동평균 변환을 수행한다. 그리고, 단계 (4)에서는 변환된 윈도우들을 저차원 변환하여 f-차원 MBR을 구성한다. 마지막으로, 단계 (5)에서는 구성한 MBR을 해당 윈도우의 시작 위치와 함께 다차원 색인에 저장한다. 이와 같은 과정을 거쳐 생성한 다차원 색인은 이후 서브시퀀스 매칭 알고리즘에 사용된다.

그런데, 데이타 시퀀스를 슬라이딩 윈도우로 나누게 되면, 너무 많은 윈도우가 생성되는 문제점이 있다. 이러한 문제점을 해결하기 위하여, FRM에서는 여러 개의 슬라이딩 윈도우를 하나의 MBR에 포함시키는 방법을 사용하였다[2]. 따라서, 제안하는 FRM-MAT에서도 여러 개의 슬라이딩 윈도우가 변환된 여러 개의 MBR을 하나의 MBR에 포함시키는 방법을 사용한다. 즉, 연속된 여러 슬라이딩 윈도우를 대표하는 MBR을 구성하고, 이 MBR을 첫번째 슬라이딩 윈도우의 시작 위치 및 마지막 슬라이딩 윈도우 시작 위치와 함께 다차원 색인에 저장하는 방법을 사용한다. 그러나, 설명의 편의상, 본 논문에서는 그림 2와 같이 각 슬라이딩 윈도우에 해당하는 MBR을 직접 저장하고 서브시퀀스 매칭을 수행하는 것으로 기술한다.

다음으로, 그림 3은 FRM-MAT의 서브시퀀스 매칭 알고리즘을 나타낸다. 우선, 단계 (1)과 단계 (2)에서는 주어진 질의 시퀀스 Q를 나누어 k-이동평균 변환한 p개의 디스조인트 윈도우 $q_i^{(k)}$를 생성한다. 다음으로, 단계 (3)~(8)에서는 각 윈도우를 사용하여 후보 서브시퀀스를 찾는 작업을 수행한다. 먼저, 단계 (4)에서는 해당 윈도우를 f-차원의 점으로 저차원 변환하고, 단계 (5)에서는 변환한 점과 $\varepsilon/\sqrt{p}$으로 범위 질의를 구성한다. 그리고, 단계 (6)에서는 앞서 구성한 범위 질의로 다차원 색인을 검색하여 $\varepsilon/\sqrt{p}$-매치하는 MBR을 찾아낸 후, 단계 (7)에서 레코드에 저장된 윈도우의 위치(offset)를 사용하여 후보 서브시퀀스들을 찾아낸다. 이와 같은 과정을 거쳐서 후보 집합을 구성한 이후에, 마지막으로 단계 (9)에서 후처리 과정을 통하여 실제 유사 서브시퀀스만을 찾는 작업을 수행한다.

3.3 DM-MAT: DualMatch with Moving Average Transform

본 절에서는 DualMatch의 서브시퀀스 매칭 방법에 다계수 이동평균 변환을 적용한 임의 계수의 이동평균 변환 서브시퀀스 매칭 방법을 설명한다. 윈도우 구성에 있어서 FRM의 이원적 접근법을 사용하는 DualMatch에서는 데이타 시퀀스를 디스조인트 윈도우로 나누고, 질의 시퀀스를 슬라이딩 윈도우로 나누는 윈도우 구성법을 사용한다. 따라서, DM-MAT는 이러한 윈도우 구성법에 다계수 이동평균 변환을 적용한 이동평균 변환

2) DualMatch와 FRM을 일반화한 GeneralMatch[14]를 사용해서도 이동평균 변환 서브시퀀스 매칭을 수행할 수 있다. 그러나, 본 논문에서는 일반적으로 널리 알려진 FRM과 FRM에 비해 성능을 크게 향상시킨 DualMatch에 대해서 알고리즘을 제시하고 성능평가를 수행한다. 그리고, GeneralMatch에 대한 적용은 향후 연구로 남겨둔다.

Procedure *FRM-MAT-SubsequenceMatching* (Query Sequence Q, Window size ω, Order k)

(1) Make $Q^{(k)}$ from Q by using k-order moving average transform;

(2) Divide $Q^{(k)}$ into disjoint windows $q_i^{(k)}(1 \le i \le p,\ p = \lfloor Len(Q^{(k)})/\omega \rfloor)$ of length ω;

(3) **for** each window $q_i^{(k)}$ **do**

(4) Transform the window to an f-dimensional point by using the lower-dimensional transformation;

(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 $S[offset-(i-1)\cdot\omega : offset-(i-1)\cdot\omega+Len(Q^{(k)})-1]$;

(8) **endfor**

(9) Do the post-processing step;

그림 3 FRM-MAT의 이동평균 변환 서브시퀀스 매칭 알고리즘

서브시퀀스 매칭 방법이다.

우선, 그림 4는 DM-MAT의 색인 구성 알고리즘을 나타낸다. 그림에서 알 수 있듯이, 데이타 시퀀스를 슬라이딩 윈도우가 아닌 디스조인트 윈도우로 나누는 점을 제외하고는 FRM-MAT의 색인 구성 알고리즘과 동일하다. 그런데, DualMatch는 데이타 시퀀스를 디스조인트 윈도우로 나누므로, 데이타 시퀀스를 슬라이딩 윈도우로 나누는 FRM에 비해 매우 적은 수(FRM의 약 $1/\omega$ 개)의 윈도우가 생성된다[8]. 따라서, 여러 MBR을 하나의 MBR에 포함시키는 FRM-MAT와는 달리, DM-MAT에서는 하나의 윈도우에 대해서 하나의 MBR을 직접 다차원 색인에 저장할 수 있다. 이러한 특징에 따라, DM-MAT에서는 FRM-MAT과 비교하여 다차원 색인에 저장되는 MBR의 크기가 작아지고, 이에 따라 색인 검색에 따른 후보 개수를 줄일 수 있다.

다음으로, 그림 5는 DM-MAT의 서브시퀀스 매칭 알고리즘을 나타낸다. 그림에서 알 수 있듯이, 질의 시퀀스를 디스조인트 윈도우가 아닌 슬라이딩 윈도우로 나누는 단계 (2)와, 이에 따라 후보 서브시퀀스를 구성하는 위치가 달라지는 단계 (7)을 제외하고는 FRM-MAT의 서브시퀀스 매칭 알고리즘과 동일하다. 그런데, DM-MAT의 서브시퀀스 매칭 알고리즘에서는 질의 시퀀스를 슬라이딩 윈도우로 나눔으로 인하여, 많은 질의 검색이 수행된다. 이러한 문제점을 해결하기 위하여 DualMatch에서와 같이, 실제로는 여러 개의 점을 포함하는 질의 MBR을 구성하여 범위 질의의 횟수를 줄이는 방법을 사용한다[8]. 그러나, 설명의 편의상, 본 논문에서는 각 점으로 직접 질의하는 것으로 기술한다.

4. 성능 평가

본 장에서는 제안한 FRM-MAT와 DM-MAT에 대한 성능 평가 결과를 설명한다. 제4.1절에서는 성능 평가를 수행한 실험 데이타와 실험 환경을 설명하고, 제4.2절에서는 실험 결과를 설명한다.

4.1 실험 데이타 및 실험 환경

제안한 방법의 우수성을 입증하기 위하여 두 가지 종류의 데이타를 사용하여 많은 실험을 수행하였다. 사용한 데이타는 하나의 긴 데이타 시퀀스로 구성된 것으로

Procedure *DM-MAT-BuildIndex*(Data Sequence S, Window size ω, Set of orders K)

(1) Divide S into disjoint windows of length ω;

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

(3) Make a set of windows $S^{(K)}[a:b]$ by using the *poly-order moving average transform* on K;

(4) Construct an f-dimensional MBR *f-D MBR* by using lower-dimensional transformations on $S^{(K)}[a:b]$;

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

(6) **endfor**

그림 4 DM-MAT의 다차원 색인 구성 알고리즘

Procedure *DM-MAT-SubsequenceMatching* (Query Sequence Q, Window size ω, Order k)

(1) Make $Q^{(k)}$ from Q by using k-order moving average transform;
(2) Divide $Q^{(k)}$ into sliding windows $Q^{(k)}[i:i+\omega-1](1 \le i \le Len(Q^{(k)})-\omega+1)$ of length ω;
(3) **for** each window $Q^{(k)}[i:i+\omega-1]$ **do**
(4) Transform the window to an f-dimensional point by using the lower-dimensional transformation;
(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 subsequence $S[offset-i+1:offset-i+\omega]$;
(8) **endfor**
(9) Do the post-processing step;

그림 5 DM-MAT의 이동평균 변환 서브시퀀스 매칭 알고리즘

서, 이는 여러 개의 데이타 시퀀스로 구성된 경우와 동일한 효과를 가진다[2,8]. 첫 번째 데이타는 기존 연구[2,4,8]에서 사용한 실제 주식 데이타로서 약 33만개의 엔트리로 구성되어 있으며, 이를 STOCK-DATA라 한다. 두 번째 데이타는 합성 데이타(synthetic data)로서 데이타 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 데이타(random walk data)이다. 이 데이타 역시 기존 연구[2,4,8]에서 사용한 것으로서 이를 WALK-DATA라 한다.

성능 평가를 위하여 순차 스캔, 제안한 방법 두 가지, 모든 계수에 대해 색인을 구성하는 방법 두 가지 등 총 다섯 가지 방법을 실험하였다.

- **SEQ-SCAN**: 순차 검색 방법으로서, 데이타베이스 전체를 한번 스캔하면서 유사 서브시퀀스를 찾는 방법이다.
- **FRM-MAT**: 제3.2절에서 제안한 방법으로, 다계수 이동평균 변환을 FRM[2]에 적용한 방법이다.
- **FRM-ORG**: 모든 이동평균 계수에 대해서 다차원 색인을 구성하고, 색인 구성과 서브시퀀스 매칭 알고리즘으로는 FRM[2]의 알고리즘을 사용한 방법이다.
- **DM-MAT**: 제3.3절에서 제안한 방법으로, 다계수 이동평균 변환을 DualMatch[8]에 적용한 방법이다.
- **DM-ORG**: 모든 이동평균 계수에 대해서 다차원 색인을 구성하고, 색인 구성과 서브시퀀스 매칭 알고리즘으로는 DualMatch[8]의 알고리즘을 사용한 방법이다.

실험을 수행한 하드웨어 플랫폼은 Intel Pentium IV 2.80 GHz CPU, 512 MB RAM, 70.0GB 하드디스크를 장착한 PC이며, 소프트웨어 플랫폼은 GNU/Linux Version 2.6.6 운영 체제이다. 다차원 색인으로는 모두 R*-트리[18]를 사용하였으며, 데이타 페이지 및 색인 페이지의 크기는 4096 바이트를 사용하였다. 그리고, 특성 추출 함수로는 DFT 변환[20]을 사용하였으며, 특성은 6개[3]를 사용하였다. 또한, 최소 질의 시퀀스 길이로 256을 사용하여, FRM-MAT 및 FRM-ORG의 윈도우 크기는 최소 질의 시퀀스 길이와 같은 256을 사용하고, DM-MAT와 DM-ORG는 FRM의 절반인 128을 사용하였다[8]. 그리고, 실험에 사용한 이동평균 계수는 2, 4, 8, 16, 32, 64, 128을 사용하였다. 즉, 계수 집합 K={2, 4, 8, 16, 32, 64, 128}로 하였다. 이에 따라, FRM-MAT 및 DM-MAT에서는 계수 집합 K에 대해서 하나의 다차원 색인을 구성하였으며, FRM-ORG 및 DM-ORG에서는 집합 K의 모든 원소인 일곱 개의 계수에 대해서 각각 다차원 색인을 구성하였다.

실험 결과로는 각 방법의 실제 수행 시간을 측정하였다. 질의 시퀀스는 데이타 시퀀스의 임의 위치(random offset)를 시작 엔트리로 하는 $Len(Q)+k-1$ 의 서브시퀀스를 추출하여 k-이동평균 변환한 시퀀스로 하였으며, 노이즈(noise) 효과를 피하기 위하여 같은 길이를 갖는 10개의 다른 질의 시퀀스에 대해서 실험한 후 평균을 취한 값을 실험 결과로 하였다. 질의에 대한 선택률은 모든 가능한 데이타 서브시퀀스 개수에 대한 질의 결과 얻은 유사 서브시퀀스 개수의 비율로서, 다음 식 (8)과 같이 정의된다.

$$\text{선택률} = \frac{\text{질의 시퀀스 } Q\text{와 유사한 서브시퀀스 개수}}{\text{데이터베이스에서 길이 } Len(Q)\text{인 모든 가능한 데이터 서브시퀀스}} \quad (8)$$

4.2 성능 시험 결과

본 절에서는 다섯 가지 방법에 대한 성능 평가 결과를 설명한다. 먼저, 실험 1)에서는 질의 시퀀스 길이를 고정하고, 여러 선택률에 대해 이동평균 계수를 달리하면서 실험을 수행하였다. 그리고, 실험 2)에서는 선택률을 고정하고, 여러 질의 시퀀스 길이에 대해 이동평균

3) DFT 변환에서는 첫 번째 복소수의 허수부 0 대신, 네 번째 복소수의 실수부를 사용하였다.

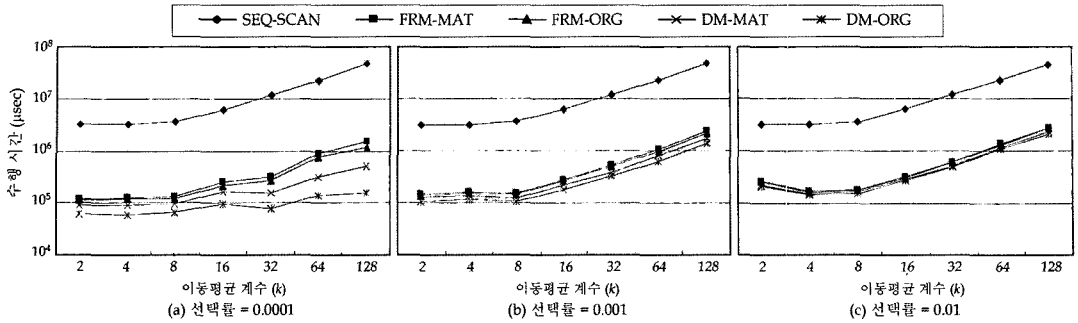


그림 6 STOCK-DATA에서 각 선택률에 대한 실험 결과(질의 시퀀스 길이 = 512)

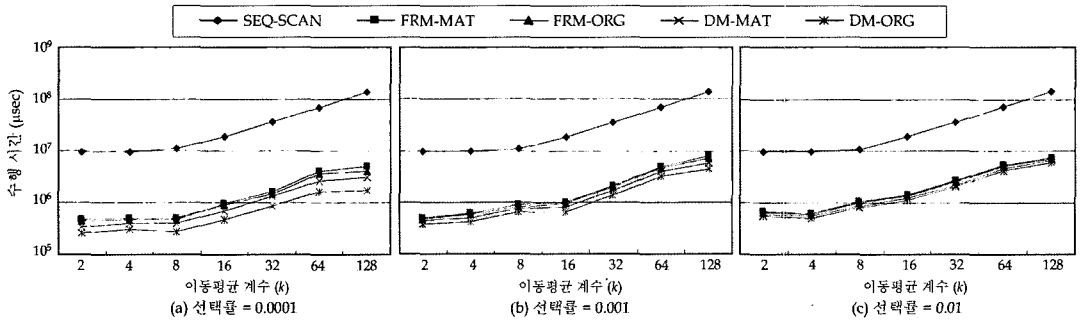


그림 7 WALK-DATA에서 각 선택률에 대한 실험 결과(질의 시퀀스 길이 = 512)

계수를 달리하면서 실험을 수행하였다. 마지막으로, 실험 3)에서는 다차원 색인을 저장하는 색인 공간 측면에서 각 방법의 장단점을 비교한다.

실험 1) 각 선택률에 대해 이동평균 계수를 달리한 경우의 실험 결과

그림 6은 STOCK-DATA에 대해 질의 시퀀스 길이를 512로 고정하고, 선택률 0.0001, 0.001, 0.01 각각에 대해 이동평균 계수를 달리하면서 수행 시간을 측정한 결과이다. 그림 6의 (a)는 선택률이 0.0001인 경우이고, (b)는 0.001, (c)는 0.01인 경우이다. 그림에서 보면, 제안한 FRM-MAT와 DM-MAT는 선택률에 관계없이 순차 스캔에 비하여 수행 시간을 크게 줄였음을 알 수 있다. 그림 6의 STOCK-DATA에 대한 실험 결과를 요약하면, FRM-MAT는 순차 스캔에 비하여 평균 22.4배 성능을 향상시키고, DM-MAT는 순차 스캔에 비하여 평균 33.8배 성능을 향상시킨 것으로 나타났다. 반면에, FRM-MAT 및 DM-MAT는 모든 계수에 대해 색인을 구성하는 FRM-ORG 및 DM-ORG에 비해서는 평균 9% 및 42%까지 각각 성능이 저하되었다. 그 이유는 색인에 저장하는 MBR 크기 측면에서 볼 때, 하나의 색인에서 모든 계수를 고려하는 FRM-MAT 및 DM-MAT의 경우가 하나의 색인에서 하나의 계수만을 고려하는 FRM-ORG 및 DM-ORG의 경우에 비하여 크기 때문이다. 그리고, DM-MAT의 성능이 FRM-MAT보다 우수한 이유는 DualMatch에서는 색인 수준 여과(index-level filtering)[4]을 수행하는 반면에 FRM에서는 그렇지 못하기 때문이다. 또한, 실험 결과에서 k 값이 작은 경우가 큰 경우보다 성능이 떨어지는 경우가 있는데, 그 이유는 이동평균 변환의 정의에 따라 계수가 작은 경우는 큰 경우보다 이웃한 각 점들 사이의 차이가 커지고, 결과적으로 이를 포함하는 MBR의 크기가 커질 수 있기 때문이다.

다음으로, 그림 7은 WALK-DATA에 대해 질의 시퀀스 길이를 512로 고정하고, 각 선택률에 대해 계수를 달리하면서 수행 시간을 측정한 결과이다. 그림을 보면, WALK-DATA의 경우도 STOCK-DATA와 마찬가지로, 제안한 FRM-MAT 및 DM-MAT의 성능이 순차 스캔에 비하여 크게 향상되었음을 알 수 있다. 그림 7의 WALK-DATA에 대한 실험 결과를 요약하면, FRM-

4) 색인 수준 여과란 색인을 검색하는 과정에서 MBR안에 포함된 점(혹은 MBR)을 여과에 사용하는 방법이다[13]. DualMatch의 경우, 질의 MBR을 사용하여 MBR안의 개별 점(혹은 개별 MBR)을 주기억 장치에 유지할 수 있으므로 색인 수준 여과가 가능한 반면에, FRM의 경우는 데이타 MBR을 사용하므로 MBR안의 개별 점(혹은 개별 MBR)을 주기억 장치 및 디스크에 유지할 수 없어 색인 수준 여과가 불가능하다.

MAT는 순차 스캔에 비하여 평균 17.8배 성능을 향상시키고, DM-MAT는 순차 스캔에 비하여 평균 22.0배 성능을 향상시킨 것으로 나타났다. 그리고, FRM-ORG 및 DM-ORG에 대한 FRM-MAT 및 DM-MAT의 성능 저하는 각각 평균 7% 및 27%에 불과한 것으로 나타났다.

실험 2) 각 질의 시퀀스에 대해 이동평균 계수를 달리한 경우의 실험 결과

그림 8은 STOCK-DATA에 대해 선택률을 0.0001로 고정하고, 질의 시퀀스 길이 256, 512, 1024 각각에 대해 이동평균 계수를 달리하면서 수행 시간을 측정한 결과이다. 그림 8의 (a)는 질의 시퀀스 길이가 256인 경우이고, (b)는 512, (c)는 1024인 경우이다. 그리고, 그림 9는 WALK-DATA에 대해 동일한 환경에서 실험한 결과이다. 그림 8과 9를 보면, 제안한 FRM-MAT와 DM- MAT는 질의 시퀀스의 길이에 관계없이 순차 스캔에 비하여 수행 시간을 크게 줄였음을 알 수 있다. 이들 실험 결과를 요약하면, 제안한 FRM-MAT 및 DM-MAT는 순차 스캔에 비해서 평균 14.8배~42.6배 이상 성능을 크게 향상시킨 것으로 나타났으며, FRM-ORG 및 DM-ORG에 대한 성능 저하는 평균 6%~98% 이하에 불과한 것으로 나타났다.

실험 3) 다차원 색인의 저장 공간에 대한 비교 결과

다음 표 2는 실험에 사용한 다섯 가지 방법의 색인 공간의 양을 나타낸다. 순차 스캔의 경우 색인을 사용하지 않으므로, 색인 사용량이 0임을 알 수 있다. 다음으로, FRM-MAT는 집합 K에 대해 하나의 색인만을 사용하는 반면에, FRM-ORG는 집합 K의 각 원소에 대해 총 일곱 개의 색인을 사용하므로 색인 저장 공간의 비율이 약 7.0배에 이름을 알 수 있다. 또한, DM-ORG의 경우도 DM-MAT에 비하여 약 7.0배의 색인 저장 공간이 더 필요함을 알 수 있다. 그리고, 이러한 차이는 이동평균 계수의 집합 K의 원소 개수가 많아질수록, 즉 다양한 계수의 이동평균 변환을 지원하면 할 수도 더욱 커지게 된다. 다시 말해서, 제안한 FRM-MAT 및 DM-MAT는 FRM-ORG 및 DM-ORG에 비해 $\frac{1}{|K|}$의 색인 개수만이 필요하고, 이에 따라 제안한 방법은 색인 공간을 $\frac{1}{|K|}$로 크게 줄일 수 있게 된다. 또한, 새로운 시계열 데이타의 추가 및 기존 데이타의 삭제 등에 따른 색인 관리의 오버헤드 역시 $\frac{1}{|K|}$로 크게 줄일 수 있게 된다.

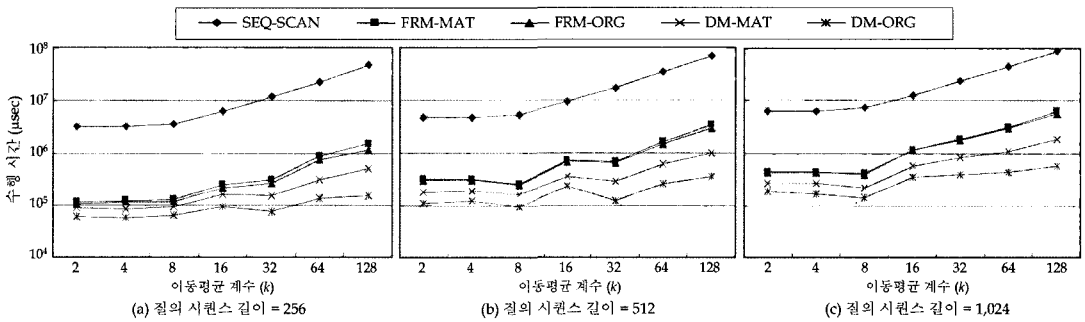


그림 8 STOCK-DATA에서 각 질의 시퀀스 길이에 대한 실험 결과(선택률 = 0.0001)

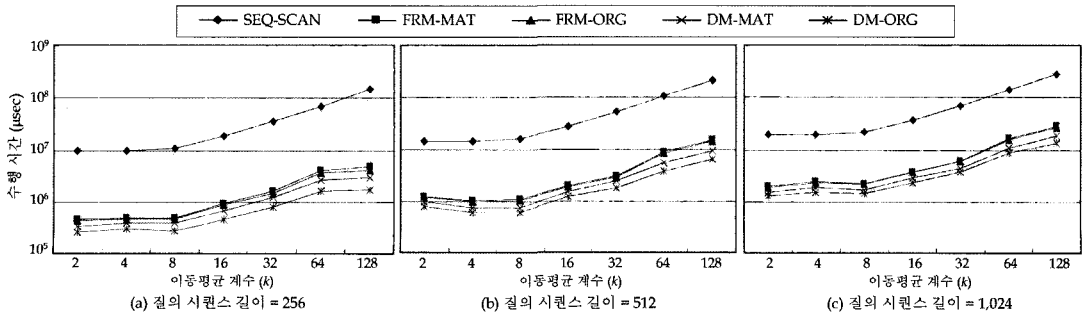


그림 9 WALK-DATA에서 각 질의 시퀀스 길이에 대한 실험 결과(선택률 = 0.0001)

표 2 다섯 가지 매칭 방법에 대한 색인 저장 공간의 비교

DATA 종류	SEQ-SCAN	FRM approach			DualMatch approach		
		FRM-MAT	FRM-ORG	FRM-ORG FRM-MAT	DM-MAT	DM-ORG	DM-ORG DM-MAT
STOCK-DATA	0 KB	218 KB	1,526 KB	7.0	198 KB	1,434 KB	7.2
WALK-DATA	0 KB	618 KB	4,254 KB	6.9	562 KB	4,158 KB	7.4

5. 결 론

본 논문에서는 하나의 색인을 사용하여 색인 저장 공간 및 관리의 오버헤드를 줄이면서도 임의 계수의 이동평균 변환을 지원하는 서브시퀀스 매칭 방법을 제안하였다. 기존의 유클리디안 거리 기반의 서브시퀀스 매칭 알고리즘[2,8]이나 임의 계수의 이동평균 변환 지원 전체 매칭 알고리즘[9]은 이동평균 변환 서브시퀀스 매칭에 그대로 적용될 수 없는 문제점이 있다. 또한, Loh 등[7]의 이동평균 변환 서브시퀀스 매칭 방법은 기존 다차원 색인의 구조와 알고리즘을 변경해야 하는 문제점과 여러 색인 생성에 따른 저장 공간의 오버헤드와 색인 관리의 오버헤드가 발생하는 문제점이 있다. 이러한 문제점을 해결하기 위하여, 본 논문에서 우선 이동평균 변환을 일반화하여 다계수 이동평균 변환의 개념을 제안하였다. 그리고, 이러한 다계수 이동평균 변환을 사용하면, 하나의 색인을 사용해서도 임의 계수에 대한 이동평균 변환 서브시퀀스 매칭을 효율적으로 수행할 수 있음을 보였다. 여기에서, 다계수 이동평균 변환이란, 각 윈도우를 특정 계수에 대해서 이동평균 변환하는 것이 아니라, 여러 계수에 대해서 이동평균 변환하여 윈도우의 집합을 구성하는 변환으로서, 단일 계수의 이동평균 변환의 정의를 여러 계수로 구성되는 집합에 대해서 확장한 것이다.

본 논문의 공헌은 다음과 같이 요약할 수 있다. 우선, 기존 연구들을 이동평균 변환 서브시퀀스 매칭에 적용할 때 나타나는 문제점을 분석하였다. 다음으로, 이동평균 변환의 개념을 확장하여 다계수 이동평균 변환의 개념을 정형적으로 정의하였다. 그리고, 다계수 이동평균 변환을 기존 서브시퀀스 매칭에 적용하는 이론적 근거를 정리로서 제시하고 증명하였다. 또한, 다계수 이동평균 변환을 기존 서브시퀀스 매칭인 FRM과 DualMatch에 적용하여, 새로운 이동평균 변환 서브시퀀스 매칭 알고리즘인 FRM-MAT와 DM-MAT를 각각 제시하였다. 마지막으로, 제안한 방법의 우수성을 데이타 종류, 선택률 범위, 질의 시퀀스 길이를 달리한 많은 실험을 통해 입증하였다. 실제 주식 데이타에 대한 실험 결과, 제안한 방법은 순차 스캔에 비해서 평균 22.4배~33.8배까지 크게 성능을 향상 시킨 것으로 나타났다. 또한, 이동평균 변환의 각 계수에 대해 모두 색인을 생성한 경우와 비교했을 때, 성능 저하는 매우 적은 반면에, 필요한 색인 공간은 크게 줄인 것으로 나타났다(일곱 개의 계수를 사용한 경우, 성능 저하는 평균 9%~42%에 불과한 반면 색인 공간은 약 1/7.0로 크게 줄인다.). 이와 같이 제안한 이동평균 변환 지원 서브시퀀스 매칭은 성능 측면에서만이 아니라 색인 공간 및 관리 측면에서 기존 방법에 비하여 우수하다고 사료된다.

본 논문에서 제안한 이동평균 변환 서브시퀀스 매칭은 정규화 변환, 스케일링 및 쉬프팅 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 확장된 혹은 일반화된 변환의 개념을 도입하고 이를 서브시퀀스 매칭에 활용한다. 따라서, 제안한 방법은 이동평균 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.

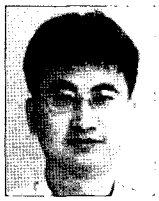
[2] Faloutsos, C., Ranganathan, M., and Manolopoulos, Y., "Fast Subsequence Matching in Time-Series Databases," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, Minneapolis, Minnesota, pp. 419-429, May 1994.

[3] 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.

[4] 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.

[5] 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.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] 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.
[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] Chatfield, C., *The Analysis of Time Series: An Introduction*, 3rd Ed., Chapman and Hall, 1984.
[17] Kendall, M., *Time-Series*, 2nd Ed., Charles Griffin and Company, 1976.
[18] 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.
[19] Berchtold, S., Bohm, C., and Kriegel, H.-P., "The Pyramid-Technique: Towards Breaking the Curse of Dimensionality," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, Seattle, Washington, pp. 142-153, June 1998.
[20] Oppenheim, A. V. and Schafer, R. W., *Digital Signal Processing*, Prentice-Hall, 1975.



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~2002월 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

김 진 호

1982년 2월 경북대학교 전자공학과 학사 1985년 2월 한국과학기술원 전산학과 석사. 1990년 2월 한국과학기술원 전산학과 박사. 1995년 8월~1996년 7월 미국 미시간 대학교 객원 교수. 2003년 2월~2004년 2월 미국 Drexel University 객원 교수. 1999년 3월~현재 한국과학기술원 첨단정보기술연구센터 연구원. 1990년 8월~현재 강원대학교 컴퓨터과학과 교수. 관심분야는 Data warehouse, OLAP, Data Mining, Real-time/Embedded Database, Main-memory database, Data Modeling, Web Database Technology