

부분 집계 근사법의 MBR-안전 성질을 이용한 효율적인 시계열 서브시퀀스 매칭

(Efficient Time-Series Subsequence Matching Using
MBR-Safe Property of Piecewise Aggregation
Approximation)

문 양 세 [†]

(Yang-Sae Moon)

요약 본 논문에서는 부분 집계 근사법(*Piecewise Aggregation Approximation: PAA*)이 MBR-안전(*MBR-safe*) 성질을 가짐을 보이고, 이를 사용한 효율적인 서브시퀀스 매칭 방법을 제안한다. MBR-안전 변환이란 고차원 MBR을 직접 변환한 저차원 MBR이 개별 고차원 시퀀스가 변환된 저차원 시퀀스를 모두 포함하는 변환을 의미한다. 이와 같은 MBR-안전 변환을 사용하면 고차원 MBR을 직접 저차원 MBR로 변환할 수 있어 유사 시퀀스 매칭에서 필요한 저차원 변환 횟수를 크게 줄일 수 있다. 또한, PAA는 계산이 간단하고 성능이 우수한 저차원 변환으로 알려져 있다. 이에 따라, 본 논문에서는 이들 두 개념의 장점을 통합하기 위하여, 기존의 PAA가 MBR-안전 성질을 가짐을 확인하고, 이를 사용하여 서브시퀀스 매칭의 성능을 개선한다. 본 논문의 공헌은 다음과 같다. 첫째, PAA 기반의 MBR 저차원 변환인 *mbrPAA*를 제안하고, *mbrPAA*가 MBR-안전함을 정형적으로 증명한다. 둘째, *mbrPAA* 기반의 새로운 서브시퀀스 매칭 방법을 제안하고, 이 방법의 정확성을 증명한다. 셋째, 서브시퀀스 매칭에서 앤트리 재사용 성질(*entry reuse property*)의 개념을 제시하고, 이 개념에 기반하여 고차원 MBR을 효율적으로 구성하는 방법을 제안한다. 넷째, 실험을 통해 *mbrPAA*의 우수성을 입증한다. 실험 결과, 제안한 *mbrPAA*는 기존 방법에 비해 저차원 MBR 구성률 평균 24.2배 빠르게 수행하고, 서브시퀀스 매칭 성능을 최대 65.9% 까지 향상시킨 것으로 나타났다.

키워드 : 시계열 데이터베이스, MBR-안전 변환, 부분 집계 근사법, 서브시퀀스 매칭

Abstract In this paper we address the *MBR-safe* property of *Piecewise Aggregation Approximation(PAA)*, and propose an efficient subsequence matching method based on the *MBR-safe PAA*. A transformation is said to be *MBR-safe* if a low-dimensional MBR to which a high-dimensional MBR is transformed by the transformation contains every individual low-dimensional sequence to which a high-dimensional sequence is transformed. Using an *MBR-safe* transformation we can reduce the number of lower-dimensional transformations required in similar sequence matching, since it transforms a high-dimensional MBR itself to a low-dimensional MBR directly. Furthermore, PAA is known as an excellent lower-dimensional transformation since its computation is very simple, and its performance is superior to other transformations. Thus, to integrate these advantages of PAA and MBR-safeness, we first formally confirm the *MBR-safe* property of PAA, and then improve subsequence matching performance using the *MBR-safe PAA*. Contributions of the paper can be summarized as follows. First, we propose a PAA-based MBR-safe transformation, called *mbrPAA*, and formally prove the *MBR-safeness* of *mbrPAA*. Second, we propose an *mbrPAA*-based subsequence matching method, and formally prove its correctness of the proposed method. Third, we

• 이 논문은 2006년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-521-D00384)

† 정 회 원 : 강원대학교 컴퓨터학부 컴퓨터과학 교수

ysmoon@kangwon.ac.kr

논문접수 : 2007년 4월 5일

심사완료 : 2007년 8월 28일

국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-521-D00384)
혹은 디지털 사본의 제작을 하기합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 데이터베이스 제34권 제6호(2007.12)

Copyright@2007 한국정보과학회

present the notion of entry reuse property, and by using the property, we propose an efficient method of constructing high-dimensional MBRs in subsequence matching. Fourth, we show the superiority of *mbrPAA* through extensive experiments. Experimental results show that, compared with the previous approach, our *mbrPAA* is 24.2 times faster in the low-dimensional MBR construction and improves subsequence matching performance by up to 65.9%.

Key words : Time-series databases, MBR-safe transformation, piecewise aggregate approximation, subsequence matching

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]. 시계열 데이터베이스에 저장된 시계열 데이터를 데이터 시퀀스라 부르며, 사용자에 의해 주어진 시퀀스를 질의 시퀀스라 부른다. 그리고, 주어진 질의 시퀀스와 유사한 데이터 시퀀스를 검색하는 방법을 **유사 시퀀스 매칭(similar sequence matching)**이라 한다[2,7]. 일반적으로, 유사 시퀀스 매칭에서는 길이 n 인 두 시퀀스 $X(= \{x_0, x_1, \dots, x_{n-1}\})$ 와 $Y(= \{y_0, y_1, \dots, y_{n-1}\})$ 의 거리가 사용자가 제시한 허용치(tolerance)인 ε 이하이면, 두 시퀀스 X 와 Y 는 유사(similar)하다고 정의한다[1,2,8,9]. 그리고, 길이 n 인 두 시퀀스 X 와 Y 의 거리 함수 $D(X, Y)$ 로는 유클리디안 거리 함수($=L_2$)를 비롯하여, 맨hattan 거리($=L_1$), 최대 거리($=L_\infty$) 등의 L_p -거리 함수($=\sqrt[p]{\sum_{i=0}^{n-1}|x_i - y_i|^p}$)를 주로 사용한다[1,2,8,10-14].

유사 시퀀스 매칭은 크게 전체 매칭(whole matching)과 서브시퀀스 매칭(subsequence matching)의 두 가지로 구분한다[2,9,10]. 전체 매칭은 질의 시퀀스와 유사한 데이터 시퀀스를 찾는 문제로서, 질의 시퀀스와 데이터 시퀀스의 길이가 동일한 특징을 갖는다[1,11]. 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 [2,4,7-10,14]. 이러한 유사 시퀀스 매칭에서는 고차원인 시퀀스를 다차원 색인에 저장하기 위하여 저차원 변환(lower-dimensional transformation)을 사용한다[1,2,7-14]. 이와 같이 저차원 변환을 사용하는 이유는 다차원 색인의 고차원 문제(high dimensionality problem) [15]를 피하고 색인의 저장 공간을 줄이기 위해서다[5,10].

본 논문에서는 우선 부분 집계 근사법(Piecewise Aggregation Approximation: PAA)이 MBR-안전(MBR-safe)[5] 성질을 가짐을 보이고, PAA를 고차원

MBR의 저차원 변환에 적용한 새로운 MBR-안전 변환을 제시한다. 어떤 저차원 변환이 MBR-안전하다 함은 고차원 MBR을 직접 변환한 저차원 MBR이 개별 고차원 시퀀스가 변환된 저차원 시퀀스를 모두 포함함을 의미한다[5]. 이와 같이 MBR-안전 변환은 여러 개의 고차원 시퀀스를 포함하는 고차원 MBR을 직접 저차원 MBR로 변환하는 방법으로, 이를 사용하면 유사 시퀀스 매칭에서의 저차원 변환 횟수를 크게 줄일 수 있다. 다음으로, PAA는 저차원 변환 방법의 하나로서, 고차원 시퀀스를 여러 구간으로 나누고, 각 구간의 평균을 해당 시퀀스의 특성(feature) 값으로 사용한다[13,16,17]. 이러한 PAA는 기존 저차원 변환 방법에 비해 계산 과정이 매우 간단하고, 성능이 우수한 것으로 알려져 있다[12,16]. 본 논문에서는 이러한 PAA가 MBR-안전 성질을 가짐에 확인하고, PAA를 고차원 MBR의 저차원 변환에 적용한 *mbrPAA*(MBR-안전 PAA)를 새로운 MBR-안전 변환으로 제시한다.

다음으로, MBR-안전 변환인 *mbrPAA*를 사용한 효율적인 서브시퀀스 매칭 방법을 제안한다. 이는 기존 서브시퀀스 매칭 방법에서 저차원 MBR 구성법을 달리함으로써 가능하다. 기존 서브시퀀스 매칭 방법에서는 여러 개의 고차원 윈도우를 저차원 변환한 후 이를 포함하는 저차원 MBR을 구성하였다. (이러한 MBR 구성법을 *mbrPAA*와 구분하여 *orgPAA*라 한다.) 반면에, 제안한 서브시퀀스 매칭 방법에서는 먼저 고차원 MBR을 구성한 후, *mbrPAA*를 사용하여 고차원 MBR을 직접 저차원 MBR로 변환한다. 이와 같이 *mbrPAA*를 사용하여 저차원 MBR 구성법을 달리함으로써, 서브시퀀스 매칭에서의 저차원 변환 횟수를 크게 줄일 수 있다. 또한, 이러한 *mbrPAA* 기반의 저차원 MBR 구성법은 기존의 모든 서브시퀀스 매칭 방법에 적용이 가능하다. 본 논문에서는 제안한 *mbrPAA* 기반 서브시퀀스 매칭 방법의 정확성을 정리로서 제시하고 증명한다.

본 논문에서는 또한 서브시퀀스 매칭에서 고차원 MBR을 구성하는 효율적인 방법을 제시한다. 제안한 *mbrPAA*를 사용하면 저차원 변환 횟수는 크게 줄일 수 있으나, 고차원 MBR 구성에 있어서 많은 비교 연산이 발생하는 문제점이 있다. 이에 따라, 본 논문에서

는 엔트리 재사용 성질(*entry reuse property*) 개념을 제시한다. 엔트리 재사용 성질이란 서브시퀀스 매칭에 서의 고차원 MBR은 시퀀스를 나눈 슬라이딩 윈도우들로 구성되고[2,10], 이에 따라 시퀀스를 구성하는 각 엔트리는 MBR의 모든 차원 구성에 사용된다는 성질이다. 그리고, 엔트리 재사용 성질을 사용하여 고차원 MBR 구성에 필요한 비교 연산의 횟수를 크게 낮추는 효율적인 방법을 제시한다. 분석 결과, 길이 n 인 m 개의 윈도우를 대상으로 할 때, 제안한 방법은 기존 방법에 비해 평균 계산 복잡도를 $O(nm)$ 에서 $O(m+n)$ 으로 크게 줄였으며, 이때 필요한 비교 연산 횟수는 엔트리 재사용 성질에 의해 $O(m+n)$ 에 불과한 것으로 나타났다. 여러 시계열 데이터에 대한 실제 실험 결과 제안한 mbrPAA는 기존 orgPAA에 비해 저차원 MBR 구성 시간을 크게 줄이고, 이를 통해 전체 서브시퀀스 매칭 성능을 향상시킨 것으로 나타났다.

본 논문의 구성은 다음과 같다. 제2장에서는 유사 시퀀스 매칭과 저차원 변환의 관련 연구를 설명한다. 제3장에서는 본 논문에서 사용하는 개념인 MBR-안전 변환과 PAA에 대해서 설명한다. 제4장에서는 PAA의 MBR-안전 버전인 mbrPAA를 제안하고, 이를 이용한 서브시퀀스 매칭 방법을 소개한다. 제5장에서는 고차원 MBR을 효율적으로 구성하는 방법을 제시한다. 제6장에서는 실험을 통해 제안한 방법의 우수성을 보인다. 마지막으로, 제7장에서 결론을 맺는다.

2. 관련 연구

제1장에서 설명한 바와 같이 유사 시퀀스 매칭은 크게 전체 매칭과 서브시퀀스 매칭의 두 가지로 구분된다 [2,5]. 그리고, 이러한 전체 매칭 및 서브시퀀스 매칭의 많은 연구에서는 유사성의 척도로서 L_p -거리 합수를 사용하였다. 또한, L_p -거리 합수가 갖는 문제점을 보완하기 위하여 다양한 전처리 기법이나 다른 거리 합수가 사용되었는데, 이를 연구에 대해서는 참고문헌 [9,19]의 이동평균 변환을, 참고문헌 [14,19]의 정규화 변환을, 참고문헌 [6,18,20]의 타임 워핑(time warping) 거리 합수를 각각 참조한다.

본 논문에서 다루는 서브 시퀀스 매칭 방법은 Faloutsos 등[2]에 의해 처음 제안되었다(저자들의 이름 첫 글자를 따서 이 방법을 *FRM*이라 한다). FRM은 데이터 시퀀스를 슬라이딩 윈도우로 나누고 질의 시퀀스를 디스조인트 윈도우로 나누는 윈도우 구성법을 사용하며, 색인 구성 알고리즘과 서브시퀀스 매칭 알고리즘으로 구성되어 있다[9]. 먼저, 색인 구성 알고리즘에서는 데이터 시퀀스를 나눈 크기 ω 의 슬라이딩 윈도우를 f -차원($\square \omega$)의 점으로 저차원 변환하여 다차원 색인인

R^* -트리[21]에 저장한다. 그런데, 데이터 시퀀스를 슬라이딩 윈도우로 나누기 때문에 너무 많은 점이 생성되는 문제점이 있다[2,14]. 이를 해결하기 위해서, FRM에서는 여러 개의 점을 포함하는 저차원 MBR을 구성하고, 이 MBR만을 R^* -트리에 저장하는 방법을 사용한다. 다음으로, 서브시퀀스 매칭 알고리즘에서는 질의 시퀀스를 나눈 크기 ω 의 디스조인트 윈도우를 f -차원 점으로 저차원 변환한 후 범위 질의를 구성한다. 그리고, R^* -트리를 검색하여 후보(*candidate*, 질의 시퀀스와 유사할 가능성이 높은 서브시퀀스) 집합을 구성한다. 이렇게 후보 집합을 구하면 *착오기각(false dismissal, 유사 시퀀스이나 착오로 인해 기각되는 서브시퀀스)*은 발생하지 않으나, 저차원 변환 사용으로 인한 *착오해답(false alarm, 후보이나 실제로는 질의 시퀀스와 유사하지 않은 서브시퀀스)*가 발생할 수 있다. 따라서, 각 후보 시퀀스에 대해서는 데이터베이스에 저장된 실제 서브시퀀스를 액세스하고, 질의 시퀀스와의 거리를 조사하여 착오해답을 제거하는 *후처리 과정(post-processing step)*을 수행한다[1,2,7-10].

DualMatch[10]와 GeneralMatch[7]는 윈도우 구성법을 달리하여 FRM의 성능을 개선한 서브시퀀스 매칭 방법들이다[9]. 우선, DualMatch에서는 윈도우 구성의 이원성(duality) 개념을 제시하고, 이원성에 기반하여 데이터 시퀀스를 디스조인트 윈도우로 나누고 질의 시퀀스를 슬라이딩 윈도우로 나누는 FRM의 이원적 접근법을 제안하였다. 이러한 이원적 윈도우 구성법에 따라, DualMatch는 다차원 색인에는 MBR 대신 개별 점을 직접 저장하고, 질의 시에 저차원 MBR을 구성하는 방법을 사용하였다. 다음으로, GeneralMatch에서는 FRM과 DualMatch에서 사용한 슬라이딩 윈도우와 디스조인트 윈도우를 일반화한 J-슬라이딩 윈도우와 J-디스조인트 윈도우 개념을 제시하고, 이를 일반화된 윈도우를 사용한 서브시퀀스 매칭 방법을 제안하였다. DualMatch와 GeneralMatch의 색인 구성 및 서브시퀀스 매칭 알고리즘은 윈도우 구성을 달리하는 것을 제외하고는 FRM의 알고리즘과 유사하다.

기존의 유사 시퀀스 매칭 방법들은 색인을 사용하기 위하여 Discrete Fourier Transform(DFT), Wavelet, PAA 등 여러 가지 저차원 변환을 사용하였다. 우선, DFT는 참고문헌 [1,2,7-9,14] 등 많은 연구에서 널리 사용되었다. 다음으로, (Haar) Wavelet 변환은 참고문헌 [10,11] 등에서, PAA는 참고문헌 [13,17] 등에서 유사 시퀀스 매칭의 저차원 변환으로 사용되었다. 이외에도, Discrete Cosine Transform(DCT)[22], Singular Value Decomposition(SVD)[12,23] 등 여러 가지 저차원 변환 방법이 제시되었다. 본 논문에서는 이러한 여

러 저차원 변환 중에서 계산이 가장 간편하고 성능이 우수한 PAA[12,16,17]를 서브시퀀스 매칭의 저차원 변환 방법으로 사용한다.

유사 시퀀스 매칭에서 필요한 저차원 변환 횟수를 줄이기 위한 연구가 참고문헌 [5]에서 수행되었다. 이 연구에서는 MBR-안전 변환의 개념을 제안하고, DFT에 대한 MBR-안전 변환을 제시하였으며, 이를 사용하면 유사 시퀀스 매칭에서의 저차원 변환 횟수를 크게 줄일 수 있음을 보였다. (MBR-안전 변환에 대한 정형적인 정의는 제3장에서 자세히 설명한다.) 그러나, 참고문헌 [5]의 연구는 1) MBR-안전 변환을 서브시퀀스 매칭에 실제 적용하지 않았고, 2) PAA에 대한 MBR-안전 변환을 다루지 않았으며, 3) 서브시퀀스 매칭에서 고차원 MBR 구성의 효율적 방법을 고려하지 않았다는 점에서 본 연구와는 차이가 있다.

3. Preliminaries

MBR-안전 PAA와 이를 이용한 서브시퀀스 매칭을 설명하기 본 논문에서 사용하는 주요 표기와 이에 대한 정의 및 의미는 표 1과 같다[5]. 설명의 편의상, 저차원 변환된 시퀀스 X^T 의 차원 f 는 고차원 시퀀스 X 의 차원 n 의 약수(factor)라 가정한다[12]. 표 1의 표기법에 따라, 우선 PAA[16,17]를 정의하면 다음과 같다.

정의 1: 차원이 n 인 시퀀스 $X = \{x_0, x_1, \dots, x_{n-1}\}$ 를 부분 집계 균사법(Piecewise Aggregation Approximation: PAA)을 사용하여 $f(\ll n)$ -차원으로 저차원 변환한 시퀀스 X^{PAA} 는 다음 공식 (1)에 의해 구해지는 시퀀스 $\{x_0^{PAA}, x_1^{PAA}, \dots, x_{f-1}^{PAA}\}$ 로 정의한다.

$$x_i^{PAA} = \frac{f}{n} \sum_{j=n(i-1)/f}^{ni/f-1} x_j \quad (1)$$

□

공식 (1)을 보면, PAA는 n/f 개의 구간에 대해서 각각 평균을 구하는 매우 간단한 과정으로 저차원 변환이 수행된다. 이에 따라, PAA는 DFT, SVD, Wavelet 변환 등에 비해 저차원 변환 과정이 매우 간단한 특징을 가지고 있다. 또한, PAA는 유사 시퀀스 매칭에서의

성능 평가 결과 DFT, SVD, Wavelet 변환 등에 비해 우수한 것으로 나타났다[12,16].

다음으로, 저차원 변환의 대상을 고차원 점이 아닌 고차원 MBR로 확장한 MBR-안전 변환의 정의는 다음과 같다[5].

정의 2: 차원이 n 인 시퀀스 X 와 같은 차원인 MBR $[L, U]$ 가 주어졌을 때, 어떤 변환 T 가 있어 $X \in [L, U]$ 이면 $X^T \in [L, U]^T$ 를 만족하면, T 는 **MBR-안전(MBR-safe)**하다고 정의한다. 즉, 변환 T 가 다음 공식 (2)를 만족하면 T 는 **MBR-안전하다고** 정의한다.

$$X \in [L, U] \Rightarrow X^T \in [L, U]^T \quad (2)$$

□

MBR-안전 개념은 유사 시퀀스 매칭에서 저차원 변환 횟수를 줄이는데 사용할 수 있다[5]. 즉, 기존 유사 시퀀스 매칭에서는 수십~수천 개의 시퀀스를 각각을 저차원 변환한 후 저차원 MBR을 구성한다[2,8,10,14]. 반면에, MBR-안전 개념을 사용하면, 수십~수천 개의 시퀀스를 포함하는 고차원 MBR을 구성한 후, 이 고차원 MBR 자체를 변환하여 저차원 MBR을 구성할 수 있다. 이러한 MBR-안전 개념을 PAA 기반의 서브시퀀스 매칭에 이용하기 위하여, 다음 제4장에서는 MBR-안전 PAA의 개념을 제시하고, 이를 이용한 서브시퀀스 매칭 방법을 제안한다.

4. MBR-안전 PAA를 사용한 서브시퀀스 매칭

본 장에서는 MBR-안전 PAA 기반의 서브시퀀스 매칭 방법을 제안한다. 제4.1절에서는 PAA 기반의 MBR-안전 변환인 mbrPAA를 제시한다. 다음으로, 제4.2절에서는 제안한 mbrPAA를 사용한 서브시퀀스 매칭 방법을 제안한다. 마지막으로, 제4.3절에서는 PAA 기반의 기존 서브시퀀스 매칭 방법과 mbrPAA 기반의 새로운 서브시퀀스 매칭 방법의 계산 복잡도를 분석한다.

4.1 mbrPAA: MBR-안전 PAA

정의 1의 PAA가 MBR-안전 성질을 만족하려면, PAA가 정의 2의 공식 (2)를 만족하여야 한다. 그런데, 지금까지는 MBR 자체에 대한 PAA가 정의된 바 없다.

표 1 주요 표기법

기호	정의/의미
X	고차원 시퀀스 혹은 고차원 원도우 ($=\{x_0, x_1, \dots, x_{n-1}\}$)
X^T	변환 T 에 의해 변환된 저차원 시퀀스 ($=\{x_0^T, x_1^T, \dots, x_{f-1}^T\}$)
$[L, U]$	고차원 MBR로서, L 은 좌하점, U 는 우상점을 나타냄 ($=\{(l_0, l_1, \dots, l_{n-1}), (u_0, u_1, \dots, u_{n-1})\}$)
$[L, U]^T = [\Lambda, \Upsilon]$	MBR $[L, U]$ 가 T 에 의해 변환된 저차원 MBR ($=\{(\lambda_0, \lambda_1, \dots, \lambda_{f-1}), (\upsilon_0, \upsilon_1, \dots, \upsilon_{f-1})\}$)
$X \in [L, U]$	시퀀스 X 가 MBR $[L, U]$ 에 포함됨을 의미함

따라서, 본 논문에서는 다음과 같이 고차원 MBR을 저차원 MBR로 변환하는 mbrPAA를 정의한다.

정의 3: 저차원 변환인 mbrPAA는 n -차원 시퀀스 X 와 n -차원 MBR $[L, U]$ 를 각각 f -차원 시퀀스 X^{mbrPAA} 와 f -차원 MBR $[L, U]^{mbrPAA}$ 로 변환하며, 이때 X^{mbrPAA} 와 $[L, U]^{mbrPAA}$ 는 다음 공식 (3)과 같이 정의한다.

$$X^{mbrPAA} = X^{PAA},$$

$$[L, U]^{mbrPAA} = [L^{mbrPAA}, U^{mbrPAA}]. \quad (3)$$

□

정의 3에 따르면 n -차원 MBR $[L, U]$ 는 mbrPAA에 의해 f -차원 MBR $[L^{mbrPAA}, U^{mbrPAA}]$ 로, 즉 MBR $[L^{PAA}, U^{PAA}]$ 로 변환된다. 이는 mbrPAA가 고차원 MBR의 좌하점과 우상점을 각각 PAA로 저차원 변환하여 저차원 MBR을 구성하는 매우 단순한 방법임을 의미한다.

다음 정리 1은 정의 3의 mbrPAA가 MBR-안전 변환임을 나타낸다.

정리 1: 시퀀스 X 와 MBR $[L, U]$ 가 주어졌을 때, $X \in [L, U]$ 이면, X 와 $[L, U]$ 를 각각 mbrPAA로 저차원 변환한 X^{mbrPAA} 와 $[L, U]^{mbrPAA}$ 사이에는 $X^{mbrPAA} \in [L, U]^{mbrPAA}$ 의 관계가 성립한다. 즉, mbrPAA는 MBR-안전하다.

증명: PAA 및 mbrPAA의 정의에 따라 쉽게 증명할 수 있다. 우선 편의상 $[L, U]^{mbrPAA}$ 를 $[\Lambda, \Upsilon]$ 이라 표기하자. 그러면, 모든 i 에 대해서 $\lambda_i \leq x_i^{mbrPAA} \leq \upsilon_i$ 가 성립하면, $X^{mbrPAA} \in [\Lambda, \Upsilon] = [L, U]^{mbrPAA}$ 가 성립하여 증명이 완료된다.

먼저, 정의 3에 의하여 $\Lambda = L^{mbrPAA} = L^{PAA}$ 이므로, PAA의 정의(정의 1)에 따라 $\lambda_i = \frac{f}{n} \sum_{j=n(i-1)/f}^{n/f-1} l_j$ 가 성립한다. 또한,

$X^{mbrPAA} = X^{PAA}$ 이므로, $x_i^{mbrPAA} = \frac{f}{n} \sum_{j=n(i-1)/f}^{n/f-1} x_j$ 가 성립한다. 그

런데, 가정에 의해 $X \in [L, U]$ 이므로, 각 j 에 대해서

$$l_j \leq x_j \text{의 관계가 성립한다. 따라서, } \sum_{j=n(i-1)/f}^{n/f-1} l_j \leq \sum_{j=n(i-1)/f}^{n/f-1} x_j$$

$$\text{및 } \frac{f}{n} \sum_{j=n(i-1)/f}^{n/f-1} l_j \leq \frac{f}{n} \sum_{j=n(i-1)/f}^{n/f-1} x_j \text{의 관계가 성립하고, 결국}$$

$\lambda_i \leq x_i^{mbrPAA}$ 이 성립한다. 동일한 방법으로, $x_i \leq \upsilon_i$ 에 의해 $x_i^{mbrPAA} \leq \upsilon_i$ 이 성립함을 보일 수 있다. 이를 종합하면, 모든 i 에 대해서 $\lambda_i \leq x_i^{mbrPAA} \leq \upsilon_i$ 가 성립하고, 정의 1에 따라 mbrPAA는 MBR-안전하다. □

다음 예제 1은 mbrPAA가 MBR-안전함을 나타내는 예이다.

예제 1: 그림 1은 4-차원 공간의 시퀀스 X 와 MBR $[L, U]$ 를 mbrPAA를 사용하여 2-차원 공간의 시퀀스 X^{mbrPAA} 와 MBR $[L, U]^{mbrPAA}$ 로 변환한 예를 나타낸다. 즉, mbrPAA를 사용한 저차원 변환에 있어서 $n = 4$ 이고, $f = 2$ 인 예제이다. 그림 1(a)를 보면, 모든 i 에 대해서 $l_i \leq x_i \leq \upsilon_i$ 이므로, $X \in [L, U]$ 이 성립함을 알 수 있다.

$$\text{그리고, 정의 3에 의해 } X^{mbrPAA} = \left\{ \frac{2}{4}(3+1), \frac{2}{4}(2+4) \right\} = \{2, 3\}$$

$$\text{으로 계산되고, 같은 방식으로 } [L, U]^{mbrPAA} = [\Lambda, \Upsilon] \text{의 } \Lambda = \left\{ \frac{2}{4}(2+0), \frac{2}{4}(1+3) \right\} = \{1, 2\}, \quad \Upsilon = \left\{ \frac{2}{4}(4+2), \frac{2}{4}(3+5) \right\} = \{3, 4\}$$

로 계산된다. 그 결과 그림 1(b)에서 보듯이, 모든 i 에 대해서 $\lambda_i \leq x_i^{mbrPAA} \leq \upsilon_i$ 가 성립하고, 결국 $X^{mbrPAA} \in [L, U]^{mbrPAA}$ 가 성립함을 알 수 있다. 즉, $X \in [L, U] \Rightarrow X^{mbrPAA} \in [L, U]^{mbrPAA}$ 의 관계가 성립하므로, 정의 2에 의해 mbrPAA는 MBR-안전함을 알 수 있다. □

4.2 mbrPAA 기반 서브시퀀스 매칭

제4.1절에서 제안한 mbrPAA는 기존 서브시퀀스 매칭 방법에 모두 적용할 수 있다. 예를 들어, FRM의 경우 색인 생성 시에 저차원 MBR을 구성하여 색인에 저장하는데, 이때 저차원 변환 방법으로 PAA 대신에

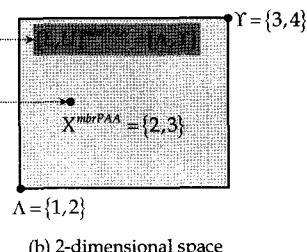
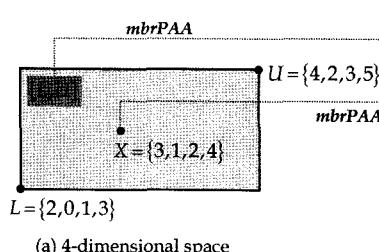


그림 1 MBR-안전 변환인 mbrPAA에 의한 저차원 변환 예제($n = 4, f = 2$)

mbrPAA를 사용할 수 있다. 즉, 데이터 시퀀스를 나눈 여러 개의 슬라이딩 윈도우를 PAA로 변환한 후 저차원 MBR을 구성하는 방법 대신에, 여러 윈도우들을 포함하는 고차원 MBR을 구성한 후 mbrPAA로 고차원 MBR 자체를 변환하여 저차원 MBR을 구성하는 방법을 사용하는 것이다. 다음으로, DualMatch의 경우 색인 검색 시에 MBR을 구성하여 범위 질의를 수행하는 데, 이때 저차원 변환 방법으로 PAA 대신 mbrPAA를 사용할 수 있다. 즉, 질의 시퀀스를 나눈 여러 슬라이딩 윈도우들로 고차원 MBR을 구성한 후 mbrPAA로 저차원 변환하여 질의 MBR을 구성하는 것이다. 이와 같이 “윈도우들을 저차원 변환한 후 MBR을 구성하는 방법”을 “MBR을 구성한 후 MBR을 저차원 변환하는 방법”으로 변경함으로써 mbrPAA를 기준 서브시퀀스 매칭에 적용할 수 있다. 본 논문의 이후 설명에서는 “윈도우들을 저차원 변환한 후 MBR을 구성하는 기준의 MBR 구성 방법”을 mbrPAA와 구분하여 orgPAA라 부른다.

앞서 설명한 바와 같이, 저차원 변환과 MBR 구성의 순서를 변경함으로써 mbrPAA를 기준 서브시퀀스 매칭 방법에 적용할 수 있다. 그런데, 이와 같은 mbrPAA 기준 서브시퀀스 매칭 방법을 사용하기 위해서는 이 방법이 착오기각을 발생하지 않음을, 즉 정확성을 증명하여야 한다. 다음 정리 2는 이러한 mbrPAA 기준 서브시퀀스 매칭 방법의 정확성을 나타낸다.

정리 2: 착오해답을 발생하지 않는 임의의 서브시퀀스 매칭 방법인 S-매칭이 주어졌고, S-매칭은 여러 개의 고차원 윈도우들을 PAA로 변환하여 저차원 MBR을 구성하는 방법인 orgPAA를 사용한다고 하자. 그러면, S-매칭에서 orgPAA를 mbrPAA로 변경한 새로운 서브시퀀스 매칭 방법은 착오기각을 발생하지 않는다.

증명: 먼저, 정리 1을 사용하여 orgPAA에 의한 MBR이 mbrPAA에 의한 MBR에 포함됨을 증명한다. 저차원 변환 대상인 고차원 윈도우 집합이 $\mathbb{X} = \{X_1, X_2, \dots, X_m\}$ 이고, 이를 윈도우를 포함하는 고차원 MBR이 $[L, U]$ 라 하자. 그리고, Y 는 MBR $[L, U]$ 내에 포함되는 임의의 윈도우라 하자 ($Y \in \mathbb{X}$ 혹은 $Y \notin \mathbb{X}$). 그러면, orgPAA에 의해 구성된 MBR(이를 MBR^{orgPAA} 라 하자)과 $[L, U]^{mbrPAA}$ 사이에는 다음 공식 (4)의 관계가 성립한다.

$$X_i^{PAA} \in MBR^{orgPAA},$$

$$Y^{PAA} \in [L, U]^{mbrPAA} \quad (\because Y^{mbrPAA} \in [L, U]^{mbrPAA}). \quad (4)$$

집합 \mathbb{X} 에 속한 모든 윈도우 X_i 와 $[L, U]$ 에 포함되는 임의의 윈도우 Y 에 대해서 공식 (4)가 성립하므로,

$MBR^{orgPAA} \subseteq [L, U]^{mbrPAA}$ 의 포함관계가 성립한다. (반대로, $MBR^{orgPAA} \supseteq [L, U]^{mbrPAA}$ 는 성립하지 않는다. 이는 $[L, U]$ 에는 포함되나 고차원 윈도우 집합 \mathbb{X} 에는 포함되지 않는 윈도우 Y , 즉 $Y \neq X_i$ 인 Y 가 있을 수 있고, $Y^{PAA} \notin MBR^{orgPAA}$ 가 될 수 있기 때문이다.) 그러면, $MBR^{orgPAA} \subseteq [L, U]^{mbrPAA}$ 의 포함관계에 의해, 임의의 윈도우 W ($W \in [L, U]$ 혹은 $W \notin [L, U]$)에 대해서 $D(W^{PAA}, MBR^{orgPAA}) \leq D(W^{mbrPAA}, [L, U]^{mbrPAA})$ 의 관계가 성립하므로, 다음 조건식 (5)가 성립한다.

$$D(W^{PAA}, MBR^{orgPAA}) \leq \epsilon \Rightarrow D(W^{mbrPAA}, [L, U]^{mbrPAA}) \leq \epsilon \quad (5)$$

조건식 (5)에서 좌변은 orgPAA 기준의 S-매칭에서 후보 집합을 구성하는 방법을 나타내고, 우변은 mbrPAA 기준의 새로운 서브시퀀스 매칭에서 후보 집합을 구성하는 방법을 나타낸다. 결국, 조건식 (5)에 의해 mbrPAA 기준의 서브시퀀스 매칭 방법은 S-매칭의 후보 집합을 포함하는 모집합(superset)을 후보 집합으로 구성하므로 착오기각이 발생하지 않는다. □

그림 2는 orgPAA와 mbrPAA를 사용한 서브시퀀스 매칭 방법들을 나타낸다. 그림 2(a)에서 보듯이, orgPAA는 고차원 윈도우 각각을 PAA로 변환한 후 저차원 MBR을 구성한다. 그리고, 이 저차원 MBR을 서브시퀀스 매칭에 사용한다. 반면에, 그림 2(b)를 보면 mbrPAA는 먼저 고차원 윈도우들을 포함하는 고차원 MBR을 구성한 후, 이를 직접 변환하여 저차원 MBR을 구성한다. 그런 다음, 마찬가지로 이 저차원 MBR을 서브시퀀스 매칭 방법에 사용한다. 그런데, 정리 2에서 증명한 바와 같이 그림의 MBR^{orgPAA} 와 $[L, U]^{mbrPAA}$ 사이에는 $MBR^{orgPAA} \subset [L, U]^{mbrPAA}$ 의 관계가 성립한다. 이는 orgPAA 기준 방법이 착오기각 없이 서브시퀀스 매칭을 수행한다면, mbrPAA 기준 방법 또한 착오기각 없이 서브시퀀스 매칭을 수행함을 의미한다.

MBR 내에 몇 개의 윈도우를 어떤 기준으로 포함시키는지에 대해서는 여러 방법이 사용될 수 있다. 예를 들어, FRM[2]에서는 서브-트레일(sub-trail) 개념을 사용하여 MBR 면적을 최소화 시키는 방법으로 하나의 MBR에 포함되는 윈도우 개수를 동적으로 결정하였다. 또한, DualMatch[10]의 경우는 질의 시퀀스의 모든 슬라이딩 윈도우들을 하나의 질의 MBR에 포함시키는 간단한 방법을 사용하였다. 그런데, 본 연구에서 다루는 고차원 MBR의 저차원 변환 문제는 MBR 내에 포함될 윈도우 개수를 결정하는 문제와는 직교적(orthogonal)

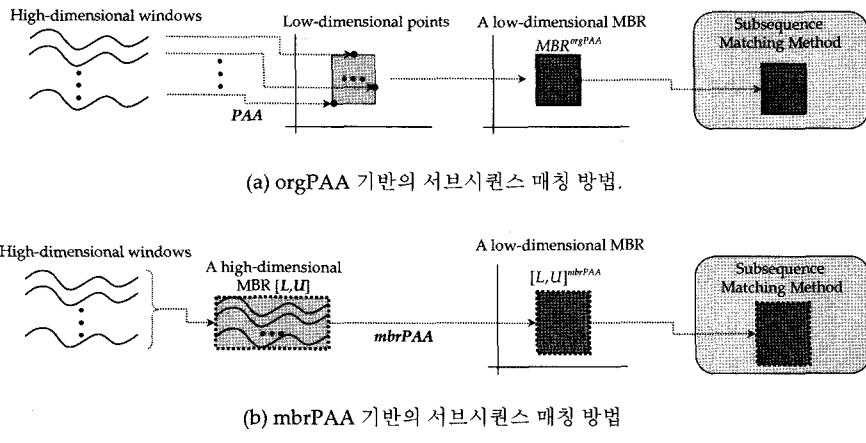


그림 2 PAA 기반의 두 가지 서브시퀀스 매칭 방법

이라 할 수 있다. 다시 말해서, 후자에 대해 더 좋은 해결책이 나오면 전자에서는 이를 사용하여 고차원 MBR을 구성하면 된다. 따라서, 본 논문에서는 FRM 혹은 DualMatch 등에서 제안된 MBR 구성법에 따라 고차원 MBR 내에 여러 개의 윈도우들을 포함된다고 가정하고 향후 논의를 전개한다.

그림 2의 과정에 의해 구성된 저차원 MBR은 실제 서브시퀀스 매칭 방법에 따라 다르게 활용될 수 있다. 먼저, FRM[2]의 경우는 저차원 MBR 구성법을 다차원 인덱스 구성에 사용하여 다차원 색인 구성 시간을 단축 시킬 수 있다. 그런데, 일반적으로 다차원 색인 구성 과정은 서브시퀀스 매칭 성능에서 고려하지 않는다. 따라서 저차원 MBR 구성법을 FRM에 적용해서 얻을 수 있는 서브시퀀스 매칭 성능 개선은 기대하기 어렵다고 볼 수 있다. 반면에, DualMatch[10]의 경우 저차원 MBR을 서브시퀀스 매칭 과정에서 질의 시퀀스를 대상으로 사용하므로, 서브시퀀스 매칭의 전체 성능을 향상 시킬 수 있다. 그리고 GeneralMatch[7]에서는 FRM과 DualMatch의 중간적인 효과, 즉 인덱스 구성과 서브시퀀스 매칭에서 수행 시간을 일부씩 단축시키는 효과를 거둘 수 있게 된다. 본 논문에서는 특정 서브시퀀스 매칭 방법을 가정하지 않고 정의 2와 그림 2를 설명하였다. 그러나, 상기 설명과 같이 실제 서브시퀀스 매칭에 따라 성능 개선 효과를 거둘 수 있는 경우와 그렇지 않은 경우가 있으며, 이에 따라서 제6장의 실험에서는 성능 개선 효과를 거둘 수 있는 DualMatch를 서브시퀀스 매칭 방법으로 사용하였다.

4.3 저차원 MBR 구성의 계산 복잡도 분석

기존 연구 [5]에서 분석하였듯이 MBR-안전 변환을 사용하면 저차원 변환 횟수를 크게 줄일 수 있다. 이에 따라 mbrPAA를 사용하면 orgPAA에 비해 저차원 변

환 횟수를 크게 줄일 수 있다. 시퀀스(윈도우)의 차원이 n 이고, 하나의 MBR에 m 개의 시퀀스가 포함된다고 하자. 그러면, orgPAA의 경우 총 m 번의 저차원 변환이 필요한 반면에, mbrPAA의 경우 단 두 번의 저차원 변환만을 필요로 한다[5]. 이에 따라, 저차원 변환의 계산 복잡도가 $O(f(n))$ 이라면, 하나의 저차원 MBR 구성을 위하여 기존 방법은 $O(m \cdot f(n))$ 의 계산 복잡도를 가지는 반면에, MBR-안전 변환은 $O(f(n))$ 의 계산 복잡도를 가진다. 그런데, 정의 1의 PAA의 경우 계산 복잡도가 $O(n)$ 임을 쉽게 알 수 있다. 따라서, 저차원 변환을 위한 계산 복잡도는 orgPAA의 경우 $O(mn)$ 이고, mbrPAA의 경우 $O(n)$ 이 된다. 이와 같이 저차원 변환 측면에서 보면 mbrPAA가 orgPAA에 비해 저차원 변환 횟수를 $1/m$ 으로 크게 줄였음을 알 수 있다.

그러나, mbrPAA를 서브시퀀스 매칭에서의 저차원 MBR 구성에 적용하기 위해서는 고차원 MBR을 구성하는 과정에서 발생하는 추가적인 비교 연산들을 고려해야 한다. 즉, orgPAA와는 달리 mbrPAA의 경우 고차원 윈도우로부터 고차원 MBR을 구성하는 데 있어서 많은 수의 비교 연산이 추가로 발생한다. 보다 정확히 이야기하면, m 개의 n -차원 윈도우들을 포함하는 n -차원 MBR을 구성하기 위해서는 총 mn 번의 비교 연산이 필요하다. (자세한 내용은 제5장을 참조한다.) 이러한 많은 수의 비교 연산은 실수의 사칙연산에 비해서는 비교적 간단하게 수행될 수 있으나, 저차원 MBR을 구성하는 실행 시간을 전체적으로 증가시키는 결과를 낳게 된다. 즉, 이는 MBR-안전 PAA의 장점인 저차원 변환 횟수의 감소가 실제 성능 개선에 영향을 미치는 효과를 약화시키게 된다. 따라서, 다음의 제5장에서는 서브시퀀스 매칭 환경에서 MBR을 구성하는 윈도우들의 엔트리 세사용 성질을 사용하여 고차원 MBR 구성에 있어서의

비교 연산 횟수를 $O(mn)$ 에서 $O(m+n)$ 으로 줄이는 효율적인 MBR 구성법을 제시한다.

5. 서브시퀀스 매칭에서 효율적인 MBR 구성법

서브시퀀스 매칭에서의 효율적인 MBR 구성법은 엔트리 재사용 성질(entry reuse property)에 기반한다. 엔트리 재사용 성질이란 서브시퀀스 매칭에서의 고차원 MBR은 시퀀스를 나눈 슬라이딩 윈도우들로 구성되고 [2,10], 이에 따라 시퀀스를 구성하는 각 엔트리는 MBR의 모든 차원 구성에 (재)사용된다는 성질이다. 다음 예제 2는 이러한 엔트리 재사용 성질을 설명한다.

예제 2: 그럼 3과 같이 시퀀스 X 가 $\{4, 2, 3, 5, 6, 8, 7, \dots\}$ 로 주어졌고, 슬라이딩 윈도우의 크기가 4라하자. 그러면, 첫 번째 슬라이딩 윈도우는 $\{4, 2, 3, 5\}$ 가 되고, 두 번째, 세 번째, 네 번째는 각각 $\{2, 3, 5, 6\}$, $\{3, 5, 6, 8\}$, $\{5, 6, 8, 7\}$ 이 된다. 이때, 첫 번째 윈도우의 첫 번째 차원 값인 4는 4-차원 MBR의 첫 번째 차원인 l_0 및 u_0 구성에 사용되고, 같은 방식으로 두 번째, 세 번째, 네 번째 윈도우의 첫 번째 차원 값인 2, 3, 5 역시 4-차원 MBR의 첫 번째 차원 구성에 사용됨을 알 수 있다. 이는 시퀀스 X 의 엔트리 대부분이 4-차원 MBR의 첫 번째 차원 구성에 사용되었음을 의미한다. 마찬가지로, 각 슬라이딩 윈도우의 두 번째 차원 값은 MBR의 두 번째 차원인 l_1 및 u_1 구성에 사용되고, 이는 시퀀스 X 의 대부분 엔트리가 4-차원 MBR의 두 번째 차원 구성에도 사용됨을 의미한다. 이를 일반화하면, 각 슬라이딩 윈도우의 i 번째 차원 값은 고차원 MBR의 i 번째 차원인 l_{i-1} 및 u_{i-1} 구성에 사용되고, 이는 시퀀스 X 의 각 엔트리가 고차원 MBR의 i 번째 차원 구성에 사용됨을 의미한다. 이와 같이 시퀀스의 각 엔

트리가 고차원 MBR의 모든 차원 구성에 반복적으로 사용되는 성질을 고차원 MBR 구성에 있어서의 엔트리 재사용 성질이라 정의한다. \square

본 논문에서는 이와 같은 엔트리 재사용 성질을 사용하여 서브시퀀스 매칭에서 고차원 MBR 구성을 위한 비교 연산 횟수를 줄인다.

우선, 일반적인 고차원 MBR 구성법을 설명하면 다음과 같다. 편의상, 슬라이딩 윈도우 크기가 n 이고, MBR 구성의 대상인 시퀀스 X 가 $\{x_0, x_1, x_2, \dots, x_{m+n-2}\}$ 라 하자. 그러면, 총 m 개의 슬라이딩 윈도우 $\{x_0, x_1, \dots, x_{n-1}\}, \{x_1, x_2, \dots, x_n\}, \dots, \{x_{m-1}, x_m, \dots, x_{m+n-2}\}$ 가 생성된다. 일반적인 n -차원 MBR 구성법은 $l_0 = \min\{x_0, x_1, \dots, x_{m-1}\}, l_1 = \min\{x_1, x_2, \dots, x_m\}, \dots, l_{n-1} = \min\{x_{n-1}, x_n, \dots, x_{m+n-2}\}$ 를 통해 좌하점 L 을 구성하고, 같은 방법으로 $u_i = \max\{x_i, x_{i+1}, \dots, x_{i+n-1}\} (0 \leq i \leq n-1)$ 을 통해 우상점 U 를 구한다. 결국, n 개 차원 각각의 l_i 및 u_i 에 대해 m 번의 비교 연산이 필요하므로, 이러한 고차원 MBR 구성법은 대략 $2 \cdot m \cdot n$ 번의 비교 연산, 즉 $O(mn)$ 의 비교 연산 횟수를 필요로 한다.

다음으로, 엔트리 재사용 성질에 기반한 고차원 MBR 구성법은 다음과 같다. 먼저, n -차원 MBR의 좌하점 L 과 우상점 U 에 대한 1-차원 좌표인 l_0 과 u_0 을 $l_0 = \min\{x_0, x_1, \dots, x_{m-1}\}$ 과 $u_0 = \max\{x_0, x_1, \dots, x_{m-1}\}$ 로 각각 구한다. 반면에, 2-차원 좌표인 l_1, u_1 은 $\min/\max\{x_1, x_2, \dots, x_m\}$ 으로 구하지 않고, 1-차원 좌표인 l_0 과 u_0 을 사용하여 다음 방법으로 구한다. 먼저 $x_0 > l_0$ 이거나 $x_0 < u_0$ 인지, 즉 x_0 이 l_0 이나 u_0 구성에 영향을 주지 않

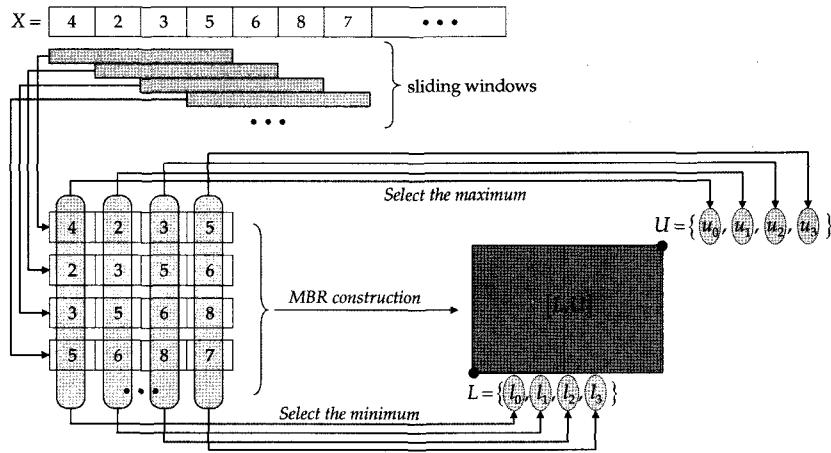


그림 3 MBR-안전 변환인 mbrPAA에 의한 저차원 변환 예제($n = 4, f = 2$)

있는지 확인한다. 만일, $x_0 > l_0$ 이라면, $\min\{x_0, x_1, \dots, x_{m-1}\} = \min\{x_1, \dots, x_{m-1}\}$ 이므로, $l_1 = \min\{l_0, x_m\}$ 으로 쉽게 구할 수 있다. 마찬가지로, $x_0 < u_0$ 이라면, $\max\{x_0, x_1, \dots, x_{m-1}\} = \max\{x_1, \dots, x_{m-1}\}$ 이므로, $u_1 = \max\{u_0, x_m\}$ 으로 쉽게 구할 수 있다. 그렇지 않고 x_0 이 l_0 이나 u_0 구성에 영향을 주었다면(즉, $x_0 = l_0$ 이거나 $x_0 = u_0$), 기존 방법과 동일하게 l_1 과 u_1 을 구한다. 마찬가지로, l_2 와 u_2 는 x_1 을 l_1 및 u_1 과 비교하여 구할 수 있다. 이러한 과정을 l_{n-1} 과 u_{n-1} 까지 계속 반복하면 m 개의 n -차원 슬라이딩 윈도우를 포함하는 n -차원 MBR을 구성할 수 있다. 이와 같은 고차원 MBR 구성법이 가능한 이유는 같은 각 엔트리가 여러 차원에 걸쳐서 영향을 미치는 엔트리 재사용 성질 때문이다.

그림 4는 지금까지 설명한 고차원 MBR 구성 알고리즘을 나타낸다. 알고리즘의 스텝 (1)은 MBR의 1-차원 값인 l_0 과 u_0 을 구하는 절차이다. 스텝 (2)~(7)은 2-차원 이후의 각 차원 값인 l_i 와 u_i 를 엔트리 재사용 성질을 사용하여 구하는 절차이다. 먼저 스텝 (3)과 (4)에서는 x_{i-1} 이 l_{i-1} 구성에 영향을 주었는지의 여부에 따라 l_i 를 다르게 구성한다. 만일 영향을 주지 않았다면(즉, $x_{i-1} > l_{i-1}$ 이면), l_i 는 스텝 (3)에서 $\min\{x_{m+i-1}, l_{i-1}\}$ 로 쉽게 구한다. 반면에, 영향을 주었다면(즉, $x_{i-1} = l_{i-1}$ 이면), l_i 는 스텝 (4)에서 $\min\{x_i, x_{i+1}, \dots, x_{i+m-1}\}$ 로 기존과 같은 방법으로 구한다. 유사한 과정을 스텝 (5)와 (6)에서 u_i 에 대해 반복한다. 스텝 (8)에서는 이와 같은 과정으로 구한 $[L, U] = [\{l_0, l_1, \dots, l_{n-1}\}, \{u_0, u_1, \dots, u_{n-1}\}]$ 을 m 개의 n -차원 슬라이딩 윈도우를 포함하는 n -차원 MBR로서 반환한다.

제안한 MBR 구성 방법의 최악의 비교연산 횟수는

```

Funcion ConstructMBR ( $X = \{x_0, x_1, \dots, x_{m+n-2}\}$ )
(1)  $l_0 := \min\{x_0, x_1, \dots, x_{m-1}\}; u_0 := \max\{x_0, x_1, \dots, x_{m-1}\}$ ;
(2) for  $i := 1$  to  $n-1$  do
(3)   if  $x_{i-1} > l_{i-1}$  then  $l_i := \min\{x_{m+i-1}, l_{i-1}\}$ ;
(4)   else  $l_i := \min\{x_i, x_{i+1}, \dots, x_{i+m-1}\}$ ;
(5)   if  $x_{i-1} < u_{i-1}$  then  $u_i := \max\{x_{m+i-1}, u_{i-1}\}$ ;
(6)   else  $u_i := \max\{x_i, x_{i+1}, \dots, x_{i+m-1}\}$ ;
(7) end-for
(8) return  $[\{l_0, l_1, \dots, l_{n-1}\}, \{u_0, u_1, \dots, u_{n-1}\}]$ ;

```

그림 4 서브시퀀스 매칭에서 고차원 MBR 구성 알고리즘

$O(mn)$ 이고, 평균 횟수는 $O(m+n)$ 이다. 먼저, 최악의 경우는 i 가 증가함에 따라, x_i 가 단조 증가(monotonic increasing)하거나 단조 감소(monotonic decreasing)하는 경우이다. 단조 증가인 경우는 항상 $x_{i-1} = l_{i-1}$ 이 만족하고, 단조 감소인 경우는 항상 $x_{i-1} = u_{i-1}$ 이 만족하여, 알고리즘의 스텝 (4) 혹은 (6)이 매번 수행되기 때문이다. 이러한 최악의 경우는 결국 n 개 차원에 대해서는 $O(m)$ 의 비교 연산을 필요로 하므로, 결국 비교 연산 횟수가 $O(mn)$ 이 된다. 다음으로, 평균 비교 연산 횟수는 그림 4의 스텝 (3)과 (5)의 실행 조건에 의해 다음과 같이 구할 수 있다. 그림 4의 알고리즘을 보면, l_i 를 구하기 위해 m 개 값을 모두 고려해야 하는 경우는 $x_{i-1} = l_{i-1}$ 인 경우로서, m 개의 엔트리 값 중에서 x_{i-1} 이 가장 작은 값을 가지는 경우이다. 마찬가지로, u_i 를 구하기 위해 m 개 값을 모두 고려해야 하는 경우는 $x_{i-1} = u_{i-1}$ 인 경우로서, m 개 엔트리 값 중에서 x_{i-1} 가 가장 큰 값을 가지는 경우이다. 즉, l_i 를 구하기 위해 m 개 엔트리 모두를 보아야 하는 확률은 x_{i-1} 이 m 개의 값 중에서 가장 작은 값을 확률로서 대략 $1/m$ 이라 할 수 있고, 마찬가지로 u_i 를 구하기 위해 m 개 엔트리 모두를 보아야 하는 확률은 x_{i-1} 이 m 개의 값 중에서 가장 큰 값을 확률로서 역시 대략 $1/m$ 이라 할 수 있다. 결국, $x_{i-1} = l_{i-1}$ 이거나 $x_{i-1} = u_{i-1}$ 일 확률은 대략 $2/m$ 이라 할 수 있다. 그리고, 이 경우에는 스텝 (4) 혹은 (6)에 의해 $O(m)$ 의 비교 연산 횟수를 나타낸다. 나머지 확률인 $(m-2)/m$ 에 대해서는 스텝 (3) 혹은 (5)에 의해 $O(1)$ 의 비교 횟수를 나타낸다. 그러므로, 스텝 (2)~(7)에서는 평균적으로

$$O(n) \left(= O\left(n \times \left(\frac{2}{m} \times O(m) + \left(\frac{m-2}{m} \times O(1)\right)\right)\right)\right)$$

의 비교 연산 횟수를 가진다. 여기에 스텝 (1)의 비교 연산 횟수인 $O(m)$ 을 더하면, 제안한 MBR 구성 알고리즘의 평균 비교 연산 횟수는 $O(m+n)$ 이 된다.

지금까지 설명한 바와 같이, 엔트리 재사용 성질을 사용하면 고차원 MBR을 $O(m+n)$ 의 평균 비교 횟수로 빠르게 구할 수 있다. 결국, 제안한 mbrPAA는 저차원 변환을 위한 평균 계산 복잡도를 orgPAA의 $O(mn)$ 에서 $O(n)$ 으로 크게 낮추었으며, mbrPAA를 사용함에 따라 추가적으로 발생하는 비교 연산의 횟수를 $O(mn)$ 에서 $O(m+n)$ 으로 낮추었다고 할 수 있다. 다음 제6장에서는 이러한 계산 복잡도와 비교 연산 횟수의 개선 효과가 실제 성능에 미치는 효과를 분석한다.

6. 성능 평가

본 장에서는 실제 실험을 통해 orgPAA 및 mbrPAA 기반 서브시퀀스 매칭 방법의 성능을 비교한다. 제6.1절

에서는 실험 데이터와 실험 환경을 소개하고, 제6.2절에서는 실험 결과를 설명한다.

6.1 실험 데이터 및 실험 환경

실험에서는 세 가지 종류의 데이터를 사용하였다. 사용한 데이터는 하나의 긴 데이터 시퀀스로 구성된 것으로서, 이는 여러 개의 데이터 시퀀스로 구성된 경우와 동일한 효과를 가진다[2,10]. 첫 번째 데이터는 기존 연구[2,7,10]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[2,7,10]에서 사용한 것으로서 이를 WALK-DATA라 한다. 세 번째 데이터는 스트리밍 시계열로서, 참고문헌 [24]에서와 같이 랜덤 워크 시리즈 y_i 에 대한 함수로서 $x_i = 100 \cdot (\sin(0.1 \cdot y_i) + 1 + i/1000000)$ 을 사용하여 생성한 100만개의 데이터이다. 여기에서, 랜덤 워크 시리즈인 y_i 로는 첫 번째 데이터인 WALK-DATA를 사용하였으며, 이 스트리밍 데이터를 SINE-DATA라 한다.

서브시퀀스 매칭 방법으로는 orgPAA 및 mbrPAA 모두에 대해 DualMatch[10]를 사용하였다. 이는 DualMatch의 경우 저차원 MBR 구성이 매 질의 시에 이루어지기 때문에 저차원 MBR 구성법 변화에 따른 성능 변화를 쉽게 확인할 수 있기 때문이다. DualMatch에서 질의 MBR을 구성했던 방법과 마찬가지로, 실험에서는 질의 시퀀스를 나눈 모든 슬라이딩 윈도우를 하나의 MBR에 포함시키는 방법을 사용하였다. 가변적인 파라미터로는 질의 시퀀스 길이(고차원 시퀀스의 차원)와 선택율[2,7,10]로서, 하나의 파라미터를 고정하고 나머지 하나를 달리하면서 실험을 수행하였다. 다음 표 2는 실험에 사용한 파라미터를 나타낸다. PAA를 사용하여 추출한 특성의 개수(저차원 시퀀스의 차원)는 8개를 사용하였다.

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. 서브시퀀스 매칭 방법인 DualMatch의 다차원 색인으로는 R*-트리[18]를 사용

하였다. 실험 결과로는 orgPAA를 사용한 경우와 mbrPAA를 사용한 경우에 대한 실제 수행 시간을 측정하였다. 특히, 실험 1)에서는 저차원 MBR 구성을 위한 수행 시간을 측정하였고, 실험 2)에서는 이를 포함한 전체 서브시퀀스 매칭 시간을 실험하였다. 또한, 실험 3)에서는 두 방법에 의해 생성된 저차원 MBR을 비교하기 위하여, MBR의 차원 길이에 대한 비교 실험을 수행하였다. 질의 시퀀스는 데이터 시퀀스의 임의 위치(random offset)를 시작 엔트리로 하는 서브시퀀스를 추출하여 사용하였으며, 노이즈(noise) 효과를 피하기 위하여 같은 길이를 갖는 10개의 다른 질의 시퀀스에 대해서 실험한 후 평균을 취한 값을 실험 결과로 하였다.

6.2 실험 결과

실험 1) 저차원 MBR 구성의 수행 시간

그림 5는 선택율을 10^{-5} 으로 고정하고, 질의 시퀀스 길이를 256, 512, 1024로 변경하면서, orgPAA와 mbrPAA의 저차원 MBR 구성 시간을 측정한 실험 결과이다. 실험에서 orgPAA의 경우, 각각의 고차원 윈도우를 저차원 변환한 후 저차원 MBR을 구성하는데 걸리는 시간을 측정하였다. 그리고, mbrPAA의 경우, 고차원 MBR을 먼저 구성한 후 이를 직접 저차원 MBR로 변환하는데 걸리는 시간을 측정하였다. 그림 5(a)의 STOCK-DATA 결과를 보면, mbrPAA가 orgPAA에 비해 수행시간을 크게 줄였음을 알 수 있다. 이는 앞서 제4장 및 제5장에서 설명한 바와 같이 mbrPAA가 orgPAA에 비해 계산 복잡도를 크게 줄이기 때문이다. 그림 5(a)을 보면, 질의 시퀀스 길이가 증가할수록 두 방법의 저차원 변환 시간이 증가함을 알 수 있다. 이는 질의 시퀀스 길이가 길수록 고려해야 하는 윈도우의 개수가 많아지기 때문이다. 그림 5(b)와 그림 5(c)의 WALK-DATA와 SINE-DATA의 경우도 STOCK-DATA와 매우 유사한 결과를 보임을 알 수 있다. 그림을 보면, orgPAA는 데이터 종류에 따른 변화가 없는 반면에, mbrPAA는 약간의 차이를 보임을 알 수 있다. 이는 orgPAA의 경우, 데이터의 종류에 관계없이 동일한 수의 저차원 변환을 수행하는 반면에, mbrPAA의 경우는 그림 4의 알고리즘에서 설명한 바와 같이 고차원 MBR을 구성하는 시간이 데이터 종류에 따라서 달라질 수 있기 때문이다. 그림 5의 결과를 종합하면, 제안한 mbrPAA는 저차원 MBR 구성 시간을 orgPAA

표 2 실험에 사용한 가변 파라미터

파라미터	기본 값	가변 값	비고
질의 시퀀스 길이	256	256, 512, 1024	윈도우 크기 = 128
선택율	10^{-5}	$10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$	

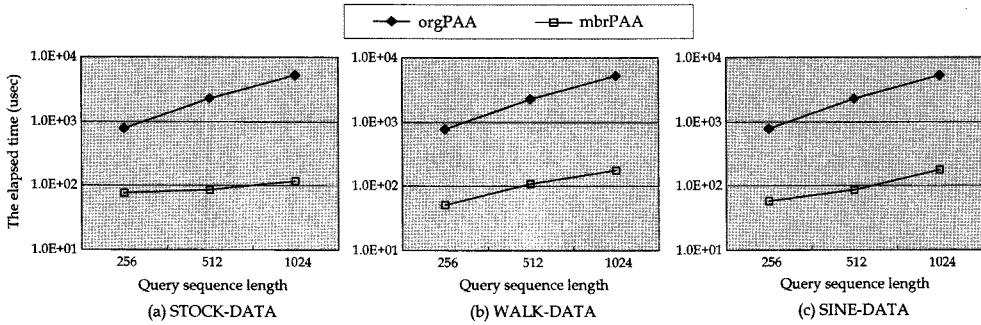


그림 5 질의 시퀀스 길이 변화에 따른 저차원 MBR 구성 시간의 비교

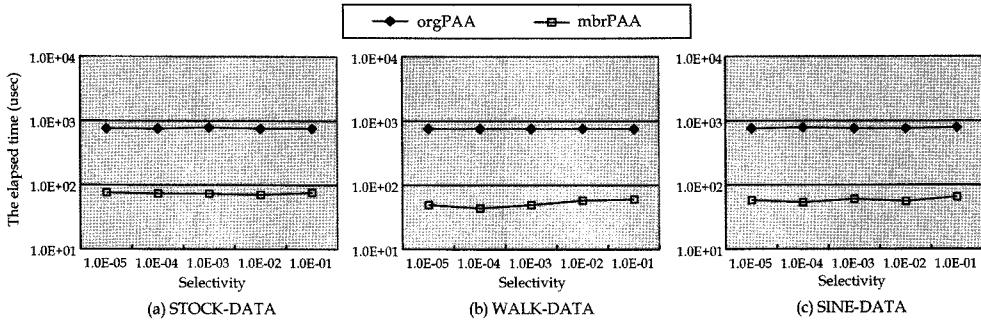


그림 6 선택율 변화에 따른 저차원 MBR 구성 시간의 비교

에 비해 평균 24.2배 줄인 것으로 나타났다.

다음으로, 그림 6은 질의 시퀀스 길이를 256으로 고정하고, 선택율을 10^{-5} 에서 10^{-1} 로 변경하면서, orgPAA와 mbrPAA의 저차원 MBR 구성 시간을 측정한 실험 결과이다. 실험 결과를 보면, 그림 5와 마찬가지로, mbrPAA는 orgPAA에 비해 저차원 MBR 구성 시간을 크게 줄였음을 알 수 있다. 그런데, orgPAA와 mbrPAA 모두 선택율을 변화에 따른 저차원 MBR 구성 시간은 큰 변화가 없음을 알 수 있다. 이는 두 방법 모두 저차원 MBR 구성 시간과 선택율과는 아무런 관계가 없기 때문이다. 즉, 선택율은 저차원 MBR 구성 이후의 색인 검색 및 후처리 과정에 영향을 미칠 뿐, 저차원 MBR 구성 자체에는 큰 영향을 미치지 않기 때문이다.

실험 2) 서브시퀀스 매칭의 수행 시간

그림 7은 선택율을 10^{-5} 으로 고정하고, 질의 시퀀스 길이를 256, 512, 1024로 변경하면서, orgPAA와 mbrPAA의 전체 서브시퀀스 매칭 시간을 측정한 실험 결과이다. 그림 7(a)의 STOCK-DATA 결과를 보면, mbrPAA가 orgPAA에 비해 서브시퀀스 매칭의 전체 성능을 향상 시켰음을 알 수 있다. 그러나, 그림 5(a)의 저차원 MBR 구성 시간과 비교해서는 성능 개선 효과가 상대적으로 작음을 알 수 있다. 이는 서브시퀀스 매

칭 과정이 저차원 MBR 구성 이외에 색인 검색, 후처리 과정 등을 포함하기 때문이다. 즉, 색인 검색과 후처리 과정에서 많은 시간이 소요되기 때문에, 전체적인 성능 개선 효과는 비교적 작게 나타나는 것이다. 다음으로, 그림 7(b)의 WALK-DATA를 보면, orgPAA와 mbrPAA의 성능 차이가 STOCK-DATA에 비해 줄었음을 알 수 있다. 이는 WALK-DATA의 경우 인접한 엔트리의 변화가 매우 적어서[10], 후보 집합의 크기가 커지고 이에 따라 후처리 과정이 오래 걸리기 때문이다. 그리고, 그림 7(c)의 SINE-DATA에서는 성능 차이가 STOCK-DATA나 WALK-DATA에 비해 큼을 알 수 있다. 이는 SINE-DATA의 경우 인접한 엔트리의 변화가 커서, 인덱싱 효과가 커지고(즉, 후보 집합의 크기가 작아지고) 이에 따라 후처리 과정이 차지하는 비중이 작아지기 때문이다. 그림 7의 실험 결과를 종합하면, 제안한 mbrPAA는 orgPAA에 비해 평균 28.5%, 최대 65.9%까지 서브시퀀스 매칭 성능을 향상 시킨 것으로 나타났다.

다음으로, 그림 8은 질의 시퀀스 길이를 256으로 고정하고, 선택율을 10^{-5} 에서 10^{-1} 로 변경하면서, orgPAA와 mbrPAA의 서브시퀀스 매칭 시간을 측정한 실험 결과이다. 실험 결과를 보면, 그림 7과 마찬가지로,

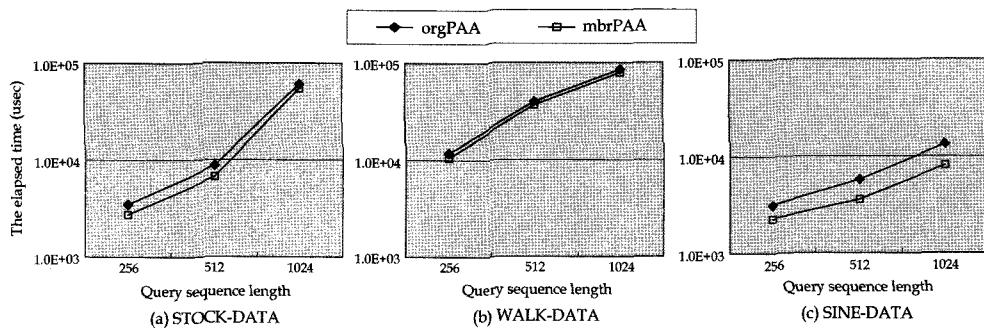


그림 7 질의 시퀀스 길이 변화에 따른 서브서비스 매칭 시간의 비교

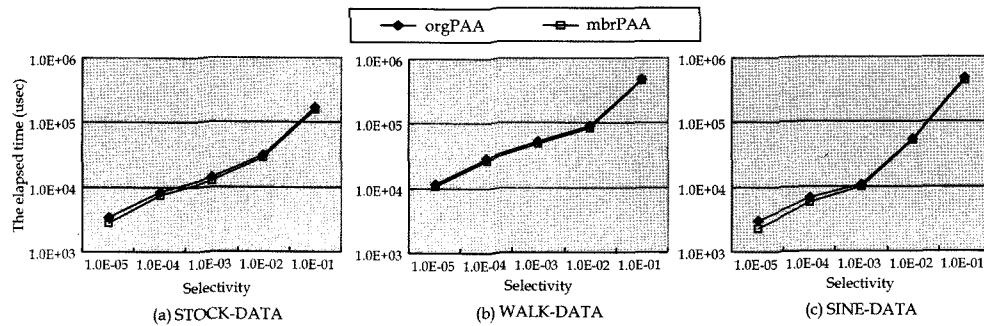


그림 8 선택율 변화에 따른 서브서비스 매칭 시간의 비교

mbrPAA는 orgPAA에 비해 서브서비스 매칭의 전체 성능을 일부 향상 시켰음을 알 수 있다. 이는 mbrPAA가 orgPAA에 비해 저차원 MBR 구성 시간을 크게 줄였기 때문이다. 그림 8의 결과를 보면, 선택율이 증가할 수록 서브서비스 매칭 시간이 증가함을 알 수 있다. 특히, 선택율이 10^{-2} 이상인 경우는 두 방법의 성능 차이가 거의 없는 것으로 나타났다(실제 실험 결과, 선택율이 10^{-2} 이상인 경우 두 방법의 성능 차이는 9% 미만이었다.). 그런데, 이는 그림 6의 저차원 MBR 구성 시간과는 다른 결과이다. 이는 선택율이 증가할 경우, 저차원 MBR 구성 시간에는 변화가 없더라도, 인덱스 검색과 후처리 과정에는 더 많은 시간이 소요되기 때문이다.

실험 3) 저차원 MBR의 차원 길이 비교

그림 9는 orgPAA와 mbrPAA에 의해 각각 생성된 저차원 MBR들의 MBR 차원 길이[5]를 비교한 실험 결과이다. MBR 차원 길이는 저차원 변환된 MBR을 이루는 각 차원의 길이를 합한 값으로서, 참고문헌 [5]에서 저차원 변환된 MBR의 단단한 정도(tightness)를 비교하기 위해 사용되었다. 이 실험을 수행한 이유는 제안한 mbrPAA이 고차원 MBR의 좌하점과 우상점만을 고려하는 반면에, orgPAA는 실제 고차원 윈도우들만을 고려하여 저차원 MBR을 구성하기 때문이다. 이

러한 이유에 의해 orgPAA에 의한 MBR과 mbrPAA에 의한 MBR은 동일하지 않게 된다. 따라서, 본 실험에서는 이들 두 방법에 의한 저차원 MBR들을 정량적으로 비교하기 위하여, 참고문헌 [5]에서 제안된 MBR 차원 길이를 비교 척도로 사용한다.

그림 9의 실험 결과를 보면, mbrPAA의 MBR 차원 길이가 orgPAA의 MBR 차원 길이보다 항상 큼을 알 수 있다. 이는 당연한 결과로서, mbrPAA가 고차원 MBR 내의 모든 가능한 가상의 윈도우들까지 고려하는 반면에, orgPAA는 실제 윈도우만을 고려하기 때문이다. 그림 9(a)의 STOCK-DATA의 경우 최소 8.7%, 최대 20.6%까지 MBR 차원 길이가 커지는 것으로 나타났으며, 그림 9(b)와 9(c)의 나머지 두 데이터도 유사한 결과를 나타내었다. 이러한 MBR 차원 길이의 증가는 더 많은 후보 집합을 생성하여 성능 저하의 한 가지 원인이 될 수 있다. 그러나, 앞서 실험 2)에서 제시한 서브서비스 매칭의 전체 성능에 있어서 mbrPAA가 orgPAA에 비해 우수한 것으로 나타났고, 이는 MBR 차원 길이의 증가가 서브서비스 매칭의 전체 성능에는 그다지 큰 영향을 미치지 않는 것으로 해석할 수 있다. 결론적으로, mbrPAA를 사용함으로써 MBR 차원 길이가 증가하기는 하나, mbrPAA를 사용함으로써 얻을 수

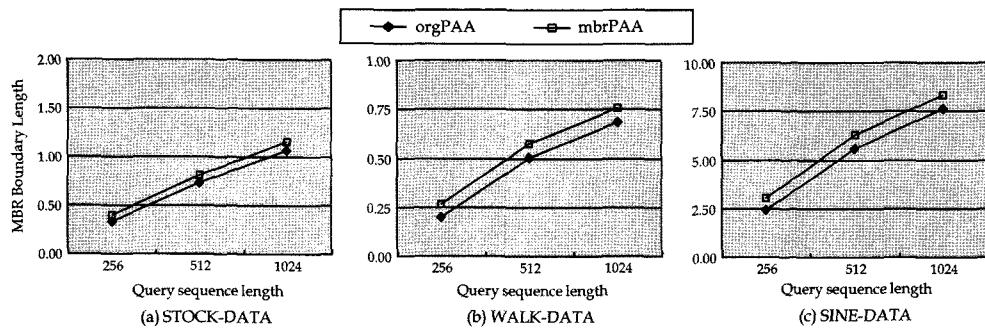


그림 9 절의 시퀀스로부터 구성된 저차원 MBR의 MBR 차원 길이 비교

있는 성능 개선 효과에 비해 크게 문제가 되지 않는 수준이라고 말할 수 있다.

4. 결 론

본 논문에서는 PAA가 MBR-안전 성질을 가짐을 보이고, 이를 사용한 효율적인 서브시퀀스 매칭 방법을 제시하였다. 유사 시퀀스 매칭에서 사용되는 PAA는 다른 변환에 의해 계산이 간단하고 성능이 우수한 것으로 알려져 있다. 그리고 MBR-안전 변환이란 고차원 MBR을 직접 변환한 저차원 MBR이 개별 고차원 시퀀스가 변환된 저차원 시퀀스를 모두 포함하는 변환으로서, 이를 사용하면 유사 시퀀스 매칭에서의 저차원 변환 횟수를 크게 줄일 수 있다. 이에 따라, 본 논문에서는 이들 두 개념의 장점을 통합하여, PAA 기반의 유사 시퀀스 매칭에서 저차원 MBR을 효율적으로 구성하는데 연구의 초점을 맞추었다. 그 결과, 기존 PAA의 MBR-안전 성질을 확인하고, PAA 기반의 새로운 MBR-안전 변환을 제시한 후, 이를 사용하여 서브시퀀스 매칭의 성능을 개선하였다.

본 논문의 공헌은 다음과 같이 1) mbrPAA를 제안하고, 2) mbrPAA 기반 서브시퀀스 매칭 방법을 제시하였으며, 3) 고차원 MBR을 효율적으로 구성하는 방법을 제안하고, 4) 실험을 통해 제안한 방법의 우수성을 입증한 네 가지로 요약할 수 있다.

- 첫째, PAA를 기반으로 고차원 MBR을 저차원 MBR로 직접 변환하는 MBR-안전 변환인 mbrPAA를 제안하였다. 제안한 mbrPAA는 고차원 MBR의 좌하점과 우상점을 각각 PAA로 저차원 변환하는 간단한 방법이다. 이러한 mbrPAA를 사용하면, 개별 시퀀스(원도우)를 변환하여 저차원 MBR을 구성하는 대신, 고차원 MBR 자체를 직접 변환하여 저차원 MBR을 구성할 수 있다. 본 논문에서는 mbrPAA가 MBR-안전함을 정리 1에서 정형적으로 증명하였다.

- 둘째, mbrPAA 기반의 서브시퀀스 매칭 방법을 제안하였다. 즉, 기존 서브시퀀스 매칭 방법에서 사용한 저차원 MBR 구성법을 mbrPAA 기반의 저차원 MBR 구성법으로 변경함으로써 새로운 서브시퀀스 매칭 방법을 제안하였다. 그리고, 이러한 mbrPAA 기반 방법이 착오기자 없이 서브시퀀스 매칭을 정확하게 수행함을 정리 2에서 증명하였다.

- 셋째, mbrPAA 기반 서브시퀀스 매칭 방법을 위한 효율적인 고차원 MBR 구성법을 제시하였다. 기존의 고차원 MBR 구성법은 비교 연산의 횟수가 많아서 mbrPAA를 사용하여 저차원 변환 횟수를 줄이는 효과를 약화시키는 문제점이 있다. 본 논문에서는 엔트리 재사용 성질의 개념을 사용하여, 길이 n 인 m 개의 윈도우를 포함하는 n -차원 MBR을 구성하는데 필요한 비교 연산의 횟수를 $O(mn)$ 에서 $O(m+n)$ 으로 줄이는 방법을 제안하였다.

- 넷째, 실험을 통해 mbrPAA의 우수성을 입증하였다. 우선, mbrPAA를 사용한 경우와 그렇지 않은 경우의 저차원 MBR 구성 시간을 비교하여, mbrPAA가 빠른 시간 내에 저차원 MBR을 구성함을 보였다. 다음으로, 기존 서브시퀀스 매칭 방법과 mbrPAA 기반 서브시퀀스 매칭 방법의 실제 수행 시간을 비교하여, mbrPAA 기반 방법의 우수성을 입증하였다.

본 논문에서 제시한 MBR-안전 PAA 개념은 유사 시퀀스 매칭뿐 아니라, 고차원 MBR을 저차원 변환해야 하는 많은 연구에 활용될 수 있다. 특히, 스트리밍 데이터 검색[24,25]이나 이미지 매칭[6,26] 분야에 활용될 수 있을 것으로 기대한다.

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] Kim, S.-W., Yoon, J., Park, S., and Won, J.-I. "Shape-based Retrieval in Time-Series Databases," *Journal of Systems and Software*, Vol. 79, No. 2, pp. 191–203, Feb. 2006.
- [4] 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.
- [5] Moon, Y.-S., "An MBR-Safe Transform for High-Dimensional MBRs in Similar Sequence Matching," In *Proc. Int'l Conf. on Database Systems for Advanced Applications (DASFAA2007)*, Bangkok, Thailand, pp. 79–90, Apr. 2007.
- [6] Keogh, E. J. et al., "LB_Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures," In *Proc. Int'l Conf. on Very Large Data Bases (VLDB)*, Seoul, Korea, pp. 882–893, Sept. 2006.
- [7] 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.
- [8] Lim, S.-H., Park, H.-J., and Kim, S.-W., "Using Multiple Indexes for Efficient Subsequence Matching in Time-Series Databases," In *Proc. of the 11th Int'l Conf. on Database Systems for Advanced Applications (DASFAA2006)*, Singapore, pp. 65–79, Apr. 2006.
- [9] Moon, Y.-S. and Kim, J., "A Single Index Approach for Time-Series Subsequence Matching that Supports Moving Average Transform of Arbitrary Order," In *Proc. of the 10th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 2006)*, Singapore, pp. 739–749, Apr. 2006.
- [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] 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.
- [12] Keogh, J., Chakrabarti, K., Pazzani, M. J., and Mehrotra, S., "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," *Knowledge and Information Systems*, Vol. 3, No. 3, pp. 263–286, Aug. 2001.
- [13] Keogh, E. J., Chu, S., and Pazzani, M. J., "Ensemble-Index: A New Approach to Indexing Large Databases," In *Proc. of the 7th Int'l Conf. on Knowledge Discovery and Data Mining*, ACM SIGKDD, San Francisco, CA, pp. 117–125, Aug. 2001.
- [14] 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.
- [15] 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.
- [16] Keogh, E. J. and Pazzani, M. J., "A Simple Dimensionality Reduction Technique for Fast Similarity Search in Large Time Series Databases," In *Proc. of the 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD 2000)*, Kyoto, Japan, pp. 122–133, Apr. 2000.
- [17] Yi, B.-K. and Faloutsos, C., "Fast Time Sequence Indexing for Arbitrary L_p Norms," In *Proc. of the 26th Int'l Conf. on Very Large Data Bases*, Cairo, Egypt, pp. 385–394, Sept. 2000.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] Hsieh, M. J., Chen, M. S., and Yu, P. S., "Integrating DCT and DWT for Approximating Cube Streams," In *Proc. of the 14th ACM Int'l Conf. on Information and Knowledge Management*, Bremen, Germany, pp. 179–186, Oct. 2005.
- [23] Korn, F., Jagadish, H. V., and Faloutsos, C., "Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences," In *Proc. of Int'l*

- Conf. on Management of Data*, ACM SIGMOD, Tucson, Arizona, pp. 289-300, June 1997.
- [24] Gao, L. and Wang, X. S., "Continually Evaluating Similarity-based Pattern Queries on a Streaming Time Series," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, Madison, Wisconsin, pp. 370-381, June 2002.
- [25] Lim, H.-S., Lee, J.-G., Lee, M.-J., Whang, K.-Y., and Song, I.-Y., "Continuous Query Processing in Data Streams Using Duality of Data and Queries," In *Proc. Int'l Conf. on Management of Data*, ACM SIGMOD, Chicago, Illinois, pp. 313-324, June 2006.
- [26] Natsev, A., Rastogi, R., and Shim, K., "WALRUS: A Similarity Retrieval Algorithm for Image Databases," *IEEE Trans. on Knowledge and Data Engineering*, Vol. 16, No. 3, pp. 301-316, Mar. 2004.

문 양 세

정보과학회논문지 : 데이터베이스
제 34 권 제 1 호 참조