시계열 데이터베이스에서 선형 추세 제거 서브시퀀스 매칭

(Linear Detrending Subsequence Matching in Time-Series Databases)

길 명 선 † 김 범 수 †
(Myeong-Seon Gil) (Bum-Soo Kim)

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

요 약 본 논문에서는 **선형 추세 제거 서브시퀀스 매칭**을 정의하고, 이를 효율적으로 수행하기 위한 인덱스 기반 해결책을 제안한다. 이를 위해, 먼저 윈도우 자체의 선형 추세가 아닌 해당 윈도우를 포함하는 서브시퀀스의 선형 추세를 제거하여 얻은 새로운 윈도우인 ***LD*-윈도우 개념**을 제시한다. 다음으로, LD-윈도우를 이용하여 제안하는 인덱스 기반 해결책의 이론적 근거인 하한 조건을 제시하고, 이를 정형적으로 증명한다. 이러한 하한 조건에 기반하여, 본 논문에서는 또한 인덱스 구성 및 서브시퀀스 매칭 알고리즘을 각각 제안한다. 마지막으로, 실험을 통해 제안하는 인덱스 기반 해결책의 우수성을 입증한다.

키워드 : 시계열 데이터베이스, 데이터 마이닝, 선형 추세 제거, 서브시퀀스 매칭

Abstract In this paper we formally define the *linear detrending subsequence matching* and propose its efficient index-based solution. To this end, we first present the notion of *LD-windows*. We eliminate the linear trend from a subsequence rather than each window itself and obtain LD-windows by dividing the subsequence into windows. Using the LD-windows we present a lower bounding theorem of the index-based solution and formally prove its correctness. Based on this lower bounding theorem, we then propose the index building and subsequence matching algorithms, respectively. Finally, we show the superiority of our index-based solution through experiments.

Key words : Time-series databases, data mining, linear detrending, subsequence matching

1. 서 론

시계열 데이터(time-series data)는 일정한 시간 별로 측정한 연속된 실수 값의 데이터로서, 그 예로는 주식 및 환율 변동 데이터, 기후 관련 데이터, 의료 측정 데이터 등이 있다[1,2]. **데이터 시퀀스**(*data sequence*)란 시계열 데이터베이스에 저장된 시계열 데이터를 말하며, **질의 시퀀스**(*query sequence*)란 사용자에 의해 주어지는 시퀀스를 말한다. 시계열 데이터베이스에서 주어진 질의 시퀀스와 유사한 데이터 시퀀스를 검색하는 방법을 **유사 시퀀스 매칭**(*similar sequence matching*)이라 한다[1,2].

유사 시퀀스 매칭에 사용되는 시계열은 왜곡(distortion)된 정보를 가질 수 있는데, 보다 직관적인 매칭 결과를 얻기 위해서는 이들 왜곡을 고려한 유사 시퀀스 매칭이 필요하다. 본 논문에서는 선형 추세 제거의 문제를 다룬다. **선형 추세**(*linear trend*)는 데이터가 전체적으로 흐르는 방향이 선형으로 나타나는 현상을 의미하며, 데이터의 방향과 가장 가까운 직선으로 표현된다.

본 논문에서는 **선형 추세 제거**(*linear detrending*)를 서브시퀀스 매칭에 적용하는 문제를 제시하고, 그 해결책을 제시한다. 전체 매칭에서 선형 추세를 제거하는 것은 질의 시퀀스와 데이터 시퀀스의 길이가 같기 때문에, 각각의 시퀀스에서 선형 추세를 제거한 뒤 유사 매칭을 수행하면 쉽게 해결할 수 있다. 하지만, 서브시퀀스 매칭 문제에서는 질의 시퀀스와 데이터 시퀀스의 길이가 다르기 때문에 고려해야 할 선형 추세가 많아지고, 그로 인해 해결 방법이 매우 복잡하다. 본 논문에서는 시계열 데이터의 선형 추세를 제거한 뒤 서브시퀀스 매칭을 수행하는 방법을 **선형 추세 제거(서브시퀀스) 매칭**(*linear detrending subsequence matching*)이라 정의한다. 본 논문에서 제안하는 선형 추세 제거 매칭은 하나의 인덱스를 사용하여 다양한 길이의 질의 시퀀스에 대한 유사

· 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.
· 이 논문은 제36회 추계학술발표회에서 '시계열 데이터베이스에서 선형 추세 제거 서브시퀀스 매칭'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 강원대학교 컴퓨터과학과
gils@kangwon.ac.kr
bskim@kangwon.ac.kr
†† 종신회원 : 강원대학교 컴퓨터과학과 교수
ysmoon@kangwon.ac.kr
(Corresponding author임)
jhkim@kangwon.ac.kr

논문접수 : 2009년 12월 22일
심사완료 : 2010년 2월 24일

Copyright©2010 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 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: 컴퓨팅의 실제 및 레터 제16권 제5호(2010.5)

매칭을 수행하는 방법이다. 이를 위해서, 서브시퀀스 매칭에서 사용하는 윈도우에 대해 선형 추세 제거한 ***LD*-윈도우**(*LD-window*)의 개념을 제시한다.

본 논문에서는 실험을 통해 제안한 인덱스 기반 해결책의 우수성을 입증하였다. 이를 위해, 인덱스 기반 해결책 및 순차 검색 해결책을 각각 구현하였으며, 이들 방법으로 실제 시계열 데이터에 대한 선형 제거 매칭을 수행하였다. 실험 결과, 본 논문에서 제안한 인덱스 기반 해결책이 순차 검색에 비해 수 배 이상 빠른 매칭을 수행하는 것으로 나타났다. 이러한 결과를 볼 때, 제안한 인덱스 기반 해결책은 선형 추세 제거 서브시퀀스 매칭을 효율적으로 수행할 수 있는 실용적인 방법이라 사료된다.

2. 관련 연구

시계열 데이터는 각 시간별로 측정한 실수 값의 시퀀스이다. 이러한 시계열 데이터의 예로는 주식 데이터, 날씨 데이터, 환율 변동 데이터 등이 있다. 유사 시퀀스 매칭은 사용자에 의해 주어진 질의 시퀀스와 시계열 데이터베이스에 저장된 데이터 시퀀스를 비교하여, 질의 시퀀스와 유사한 데이터 시퀀스를 찾는 작업이다[2]. 본 논문에서는 두 시퀀스 사이의 거리가 사용자가 제시한 허용치(tolerance)인 ε 이하이면 시퀀스 X와 Y는 유사(similar)하다고 정의[2]한 유클리디안 거리에 기반한 유사 모델[2]을 사용하여 유사 시퀀스 매칭을 수행한다.

기존의 유사 시퀀스 매칭은 크게 전체 매칭(whole matching)과 서브시퀀스 매칭(subsequence matching)으로 구분한다[2]. 전체 매칭은 질의 시퀀스와 데이터 시퀀스의 길이가 같을 때의 유사 매칭 문제이며[1], 서브시퀀스 매칭은 임의 길이의 질의 시퀀스와 데이터 시퀀스에 포함된 서브시퀀스들의 유사 매칭 문제이다. 서브시퀀스 매칭은 전체 매칭을 일반화한 것으로, 전체 매칭에 비하여 다양한 분야에 응용될 수 있다[2-6].

Faloutsos 등[2]은 전체 매칭을 일반화한 서브시퀀스 매칭 문제를 처음 소개하고, 이의 해결책(FRM)을 제시하였다. FRM은 인덱스 구성 알고리즘과 서브시퀀스 매칭 알고리즘으로 구성된다. 먼저, 인덱스 구성 알고리즘에서는 데이터 시퀀스를 나눈 슬라이딩 윈도우를 특성 추출 함수를 통해 f-차원의 점으로 변환하고, 이를 여러 개의 점을 포함하는 MBR(minimum bounding rectangle)을 구성하여 다차원 인덱스인 R*-트리에 저장한다. 다음으로, FRM의 서브시퀀스 매칭 알고리즘에서는 질의 시퀀스를 디스조인트 윈도우로 나누고, 특성 추출 함수를 통해 f-차원의 점으로 변환한 뒤, 이 점과 범위 질의를 구성한다. 그리고 R*-트리를 검색하여 찾아진 MBR이 나타내는 서브시퀀스들로 후보 집합을 구성한다. 마지막으로, 후처리 과정을 통하여 착오해답을 제거하고 유사 서브시퀀스만을 찾는다.

시계열에 나타날 수 있는 왜곡에는 크게 위치 이동(offset translation), 진폭 조정(amplitude scaling), 잡음(noise), 선형 추세(linear trend)의 네 가지가 있다[3]. 선형 추세를 제외한 시계열의 왜곡 문제는 기존에 많은 연구가 되어왔다. 먼저, 위치 이동과 진폭 조정 문제는 정규화 변환을 이용하여 해결할 수 있는데, 그 해결책은 Loh 등[5]과 Moon 등[7]을 참조한다. 다음으로, 잡음을 제거하는 문제는 Moon 등[6]과 Loh 등[4]이 이동평균 변환을 이용하여 해결하는 방법을 제안하였다.

그러나, 선형 추세를 갖는 시계열 데이터에 대한 유사 매칭 문제는 아직 해결된 바 없다. 이 문제는 전체 매칭 방법으로는 쉽게 해결이 가능하지만, 서브시퀀스 매칭 방법으로는 쉽게 해결되지 않는데, 그 이유는 다음 제 3.1절에서 자세한 설명을 하기로 한다.

3. 선형 추세 제거 서브시퀀스 매칭

3.1 문제 정의

추세는 데이터가 전체적으로 어떤 방향을 가리키는지를 뜻하는 말로써, 시계열 전체 혹은 일부에 흔히 나타나는 현상이다. 이러한 추세의 형태가 선형(직선)으로 나타날 때 이를 **선형 추세**(*linear trend*)라 한다. 이와 같은 선형 추세는 경제·통계학 분야에서 예측과 분석에 매우 유용하게 사용한다[8].

본 논문에서는 선형 추세를 시계열 데이터에 나타나는 왜곡(distortion) 중 하나로 보며, 선형 추세는 해당 데이터와 가장 근접한 직선으로 표현할 수 있다[3]. 이러한 선형 추세를 나타내는 직선을 구하는 방법은 여러 가지가 있으며, 본 논문에서는 가장 대표적인 방법인 **최소 제곱법**(*least square method*)을 사용한다.

선형 추세 제거(*linear detrending*)는 원본 시퀀스에서 해당 선형 추세를 제거하여 새로운 시퀀스를 생성하는 방법을 의미하며, 이러한 시퀀스를 다음과 같이 정의한다.

정의 1. 시퀀스 $Y=\{Y[1],Y[2],\cdots, Y[n]\}$와 이의 선형 추세 $g(k)=ak+\beta$가 주어졌을 때, 다음 식 (1)로 구해지는 시퀀스 $\overline{Y}=\{\overline{Y}[1],\overline{Y}[2],...,\overline{Y}[n]\}$를 Y의 **선형 추세 제거 시퀀스** 혹은 간단히 ***LD*-시퀀스**(*LD-sequence*)라 정의한다.

$$\overline{Y}[k]=Y[k]-g(k),\ \text{where}\ k=1,2,...,n. \quad (1)$$

그림 1은 선형 추세를 제거하기 전과 후의 시퀀스를 비교한 예제이다. 그림에서 왼편은 선형 추세를 제거하기 전의 원본 질의 시퀀스 Q와 원본 데이터 시퀀스 S를 나타낸다. 반면에 오른편은 Q와 S 각각에서 선형 추

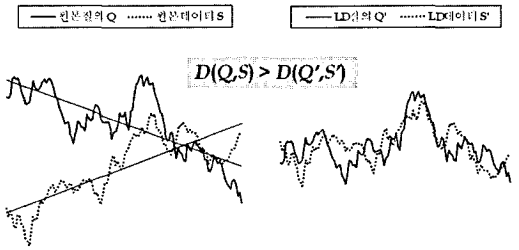


그림 1 선형 추세 제거 전과 후의 시퀀스 간 거리 비교

세가 제거된 *LD*-시퀀스 $\overline{Q}$와 $\overline{S}$를 나타낸다. 그림에서 보듯이, 선형 추세가 제거되기 전인 왼편의 두 시퀀스 Q와 S 사이에는 상당한 거리가 존재하고, 이는 궁극적으로 두 시퀀스가 유사하지 않은 것으로 판단되는 결과를 초래한다. 반면에, 선형 추세가 제거된 오른편의 *LD*-시퀀스 $\overline{Q}$와 $\overline{S}$의 거리는 매우 작고, 이는 궁극적으로 두 시퀀스가 유사하다고 판단하는 근거가 될 수 있다. 결국, 선형 추세 제거 이전에는 유사하지 않았더라도 선형 추세 제거 후에는 유사해지거나, 반대로 선형 추세 제거 이전에는 유사한 것으로 보였으나 선형 추세 제거 후에는 유사하지 않는 경우가 종종 발생할 수 있다. 이처럼 선형 추세 제거는 시계열 데이터에 나타나는 증가 혹은 감소 추세 때문에 알 수 없었던 시계열 데이터의 변동량의 유사성을 파악하는 데에 매우 유용하다. 본 논문에서는 이러한 관찰에 근거하여, 시계열 매칭에 있어서 선형 추세를 제거한 이후에 시퀀스 간의 유사성을 판단하는 문제를 다루고자 한다.

선형 추세 제거는 서브시퀀스 매칭에서는 해결이 매우 어려운 문제이다. 이를 좀 더 자세히 설명하면 다음과 같다. 먼저, 질의 시퀀스의 길이가 달라짐에 따라 이에 대응하는 서브시퀀스들이 달라지는데, 이들 서브시퀀스들의 선형 추세는 제각기 다르게 된다. 또한, 동일한 길이의 질의 시퀀스라 하더라도 해당 서브시퀀스가 데이터 시퀀스 내의 어느 위치에 나타나느냐에 따라 선형 추세가 제각기 달라진다. 결국, 질의 시퀀스 길이 및 서브시퀀스의 위치에 따라서 제각기 다른 선형 추세를 고려해야 하는 어려움이 나타나게 되고, 이로 인해 기존의 서브시퀀스 매칭 알고리즘을 선형 추세 제거에 그대로 사용할 수 없게 된다. 본 논문에서는 이러한 선형 추세 제거 서브시퀀스 매칭 문제를 해결하고자 하며, 이 문제를 다음과 같이 정형적으로 정의한다.

정의 2. 길이가 같은 두 시퀀스 X와 Y의 *LD*-시퀀스를 각각 $\overline{X}$와 $\overline{Y}$라 할 때, $\overline{X}$와 $\overline{Y}$의 유클리디안 거리 $D(\overline{Q},\overline{Y})$가 허용치 ε 이하이면 두 시퀀스 X와 Y는 ***LD*-유사**(*LD-similar*)하다고 정의한다.

정의 3. 데이터 시퀀스 S, 질의 시퀀스 Q, 허용치 ε이 주어졌을 때, Q와 *LD*-유사한 S의 서브시퀀스 $S[i:j]$ 즉, $D(\overline{Q},\overline{S[i:j]}) \le \varepsilon$ 를 만족하는 $S[i:j]$를 찾는 문제를 **선형 추세 제거 서브시퀀스 매칭**(*linear detrending subsequence matching*) 혹은 간단히 **선형 추세 제거 매칭**이라고 정의한다.

3.2 인덱스 기반 해결책

선형 추세 제거 매칭을 위해서는 데이터 및 질의 시퀀스의 선형 추세를 제거해야 하는데, 이 과정에서 데이터 시퀀스를 나눈 각 윈도우는 여러 윈도우로 매핑된다. 그 이유는 서브시퀀스 길이와 위치에 따라 달라지는 선형 추세를 각 윈도우에 모두에 적용해야 하기 때문이다. 다시 말해서, 서브시퀀스 길이는 질의 시퀀스 길이에 따라 달라지고, 데이터 시퀀스 내 어느 위치에 나타나는지도 고려해야 하기 때문이다. 이에 따라, 데이터 시퀀스 S를 나눈 윈도우 각각의 선형 추세를 제거하는 것이 아니라, 윈도우 $S[a:b]$를 포함하는 모든 서브시퀀스의 선형 추세를 $S[a:b]$에서 제거해야 하고, 이로 인해 윈도우 하나가 여러 윈도우로 매핑되는 것이다.

데이터 시퀀스를 나눈 윈도우들을 대상으로 선형 추세를 제거하는데 있어서, 윈도우 자체의 선형 추세가 아닌 윈도우를 포함하는 서브시퀀스의 선형 추세를 제거해야 한다. 이를 다루기 위해 본 논문에서는 다음 정의와 같이 *LD*-윈도우 개념을 제시한다.

정의 4. 시퀀스 S의 서브시퀀스가 $S[i:j]$이고, $S[i:j]$의 선형 추세 함수가 $g(x)$이며, $S[i:j]$에 포함된 윈도우가 $S[a:b]$이라 할 때, 윈도우 $S[a:b]$에서 선형 추세 $g(x)$를 제거하여 얻은, 즉 $S[k]$-$g[k]$를 엔트리 (k=a,...,b)로 하여 얻은 새로운 윈도우를 $S[i:j]$에 대한 $S[a:b]$의 ***LD*-윈도우**(*LD-window*)라 정의하고, 이를 $\overline{S_{[i:j]}[a:b]}$로 표기한다.

기존 서브시퀀스 매칭 방법[2,7]과 마찬가지로 제안하는 인덱스 기반 해결책에서도 각 윈도우를 저차원 변환하여 다차원 인덱스에 저장하는 방법을 사용한다. 그러나 기존 방법과는 달리, 제안하는 방법에서는 데이터 시퀀스를 나눈 하나의 윈도우를 저차원 공간의 하나의 점이 아닌 여러 점을 포함하는 하나의 MBR로 매핑한다. 이는 선형 추세 제거 매칭에서는 데이터 시퀀스를 나눈 하나의 윈도우에 대해, 서브시퀀스 길이 및 위치에 따라 여러 *LD*-윈도우들이 생성되기 때문이다. 이와 같이 하나의 윈도우가 (1) 여러 *LD*-윈도우로 매핑되고, (2) 이들 *LD*-윈도우가 저차원 변환된 후, (3) 변환된 저차원 점들을 포함하는 MBR을 다음과 같이 정의한다.

정의 5. 데이터 시퀀스 S의 한 윈도우를 s라 하고, 서브시퀀스의 모든 가능한 길이와 위치를 고려하여 생

성한 s의 LD-윈도우 $\overline{S}$들로 구성된 집합을 S라 하며, 저차원 변환 함수를 $T(x)$라 했을 때, S에 포함된 LD-윈도우 $\overline{s}(\in S)$들을 저차원 변환한 점 $T(\overline{s})$들을 모두 포함하는 MBR을 LD-MBR이라 정의하며, 이를 $\mathbb{M}(T(\overline{s}))$라 표기한다.

그림 2는 하나의 윈도우 $S[a:b]$에 대해서 LD-MBR을 구성한 예를 나타낸다. 그림에서 보듯이, 윈도우 $S[a:b]$에 대해서 먼저 여러 LD-윈도우를 생성하고, 이들 각 LD-윈도우를 저차원 변환한 후, 저차원 공간의 점들로 LD-MBR을 구성함을 알 수 있다.

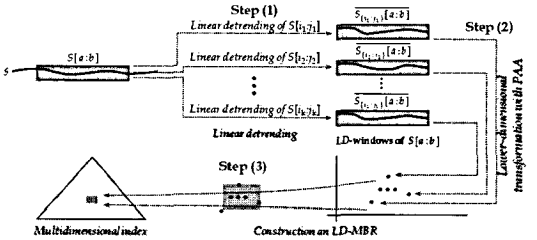


그림 2 하나의 윈도우가 LD-MBR로 매핑되는 예제

본 논문에서 제안하는 인덱스 기반 해결책은 FRM[2]의 보조정리를 근거로 얻은 다음의 정리 1에 기반을 두어 수행된다.

정리 1. 질의 시퀀스 Q와 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$의 LD-시퀀스를 각각 $\overline{Q}$와 $\overline{S[i:j]}$라 하고, $\overline{Q}$와 $\overline{S[i:j]}$를 나눈 p개의 디스조인트 윈도우(LD-윈도우)들을 각각 $\overline{q_1}$,..., $\overline{q_p}$과 $\overline{s_1}$,..., $\overline{s_p}$라 하며, 저차원 변환 함수를 $T(x)$라 하자. 이때, $\overline{Q}$와 $\overline{S[i:j]}$의 거리가 ε 이하라면(LD-유사하다면), 적어도 하나 이상의 LD-윈도우 쌍 $(\overline{q_k}, \overline{s_k})$과 이들을 저차원 변환한 쌍 $\left(T(\overline{q_k}), T(\overline{s_k})\right)$ 및 $\left(T(\overline{q_k}), \mathbb{M}(T(\mathbb{S}_k))\right)$의 거리는 모두 $\varepsilon / \sqrt{p}$ 이하이다. 즉, 다음 식 (2)이 성립한다.

$$D(\overline{Q}, \overline{S[i:j]}) \le \varepsilon \Rightarrow \bigvee_{k=1}^{p} D\left(T(\overline{q_k}), \mathbb{M}(T(\mathbb{S}_k))\right) \le \varepsilon/\sqrt{p} \quad (2)$$

여기서, $\mathbb{S}_k$는 $\overline{s_k}$를 포함하는 LD-윈도우 집합($\overline{s_k} \in \mathbb{S}_k$)이다. (증명: 참고문헌 [7]의 정규화 매칭 내용 참조)

선형 추세 제거 매칭의 인덱스 기반 해결책은 크게 인덱스 구성과 서브시퀀스 매칭으로 구성된다. 그림 3은 인덱스 구성 알고리즘을 나타낸다. 인덱스 구성에서는 데이터 시퀀스를 나눈 윈도우 각각에 대해, 먼저 LD-윈도우들을 생성하고, 이를 저차원 변환을 통해 저차원 공간의 점들로 변환하며, 저차원 공간의 점들을 포함하는 하나의 LD-MBR을 구성한 후 이 LD-MBR을 다차원 인덱스에 저장한다.

Procedure *BuildIndex* (Data sequence S, Window size ω, Query lengths l_{min} l_{max})
(1) Divide S into windows of size ω;
  // *sliding* windows for FRM; *disjoint* windows for DualMatch
(2) **for each** window $S[a:b]$ **do**
(3) Make an f-dimensional MBR $\mathbf{M}$ which is initially empty;
(4) **for each** query length $l \in [l_{min}, l_{max}]$ **do**
(5) **for each** subsequence $S[i:j]$ of length l that includes $S[a:b]$ **do**
(6) (Step ①) Compute a straight line of $S[i:j]$ based on LSM;
(7) (Step ②) Obtain the LD-window $\overline{S_{i,j}}[a:b]$; // linear detrending
(8) (Step ③) Transform that window to an f-dimensional point and include it into $\mathbf{M}$;
(9) **end for**
(10) **end for**
(11) Make a record < $\mathbf{M}$, offset = a> for $S[a:b]$, and store it into the index;
(12) **end for**

그림 3 인덱스 구성 알고리즘 BuildIndex

다음으로 인덱스 구성 과정에서 다차원 인덱스가 생성되면 이를 이용하여 서브시퀀스 매칭을 수행한다. 서브시퀀스 매칭에서는 구성된 다차원 인덱스를 검색하여 주어진 질의 시퀀스와 LD-유사할 가능성이 높은 후보 서브시퀀스들을 먼저 찾아낸 후, 이들 후보들에 대한 후처리 과정을 통해 실제로 LD-유사한 서브시퀀스만을 식별한다. 그림 4는 이러한 서브시퀀스 매칭 알고리즘을 나타낸다.

Procedure *SubsequenceMatching* (Query sequence Q, Windows size ω, Tolerance ε)
(1) Obtain $\overline{Q}$ from Q by eliminating the linear trend;
(2) Divide $\overline{Q}$ into windows of size ω;
  // *disjoint* windows for FRM, *sliding* windows for DualMatch
(3) **for each** window $\overline{Q}[a':b']$ **do**
(4) Transform that window to an f-dimensional point;
  // lower-dimensional transformation
(5) Construct a range query using that point and $\varepsilon/\sqrt{p}$;
  // p=number of included windows in Q
(6) Evaluate the query on the index and find the records of the form $\langle \mathbf{M}, a \rangle$;
(7) Include in the candidate set the subsequences $S[i:j]$ obtained from $\langle \mathbf{M}, a \rangle$;
(8) **end for**
(9) Perform the post-processing step to eliminate false alarms;

그림 4 서브시퀀스 매칭 알고리즘 SubsequenceMatching

4. 실험 결과

실험에는 총 세 가지 종류의 데이터를 사용하였다. KOS_DATA는 심전도 측정 데이터, TICK_DATA는 환율 측정 데이터, ERP_DATA는 세율 관련 데이터이며, 이는 기존 연구[3]에 사용된 실제 시계열 데이터이다[2,9]. 이 세 가지 데이터는 각각 10만개의 엔트리를 갖는다.

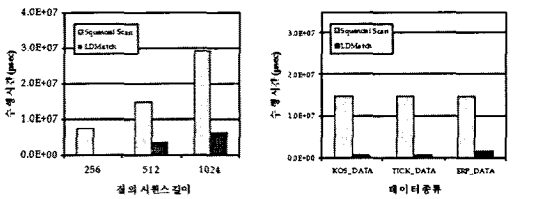
하드웨어 플랫폼은 UltraSPARC IIIi CPU 1.34GHz의 SUN Ultra 25이고, 소프트웨어 플랫폼은 Solaris 10 운영체제이다. 다차원 인덱스는 R*-트리를 사용하였다. 모든 실험에서는 하나의 인덱스를 생성하였으며, 최소 질의 시퀀스 길이는 256을 사용하고, 윈도우 크기 역시

256으로 설정하였다. 저차원 변환을 위한 특성 추출 함수는 PAA를 사용하였고, 특성은 8개를 사용하였다.

실험 결과로는 실제 수행 시간을 측정하였다. 질의 시퀀스는 데이터 시퀀스의 임의 위치(random offset)를 시작으로 하는 *Len*(*Q*)의 서브시퀀스를 사용하였으며, 노이즈 효과를 피하기 위해 같은 길이의 20개의 다른 질의 시퀀스에 대해서 실험한 후 평균값을 결과로 하였다.

제안한 선형 추세 제거 서브시퀀스 매칭 방법의 성능이 우수함을 보이기 위해, 순차 검색과 성능을 비교한다. 실험의 종류는 총 두 가지이다. 첫 번째는 데이터 길이는 10만 개, 윈도우 크기는 256으로 고정하고 질의 시퀀스의 길이에 변화를 주어 유사 매칭 수행 시간을 측정하였고, 두 번째는 데이터 길이는 10만 개, 윈도우 크기는 256, 질의 시퀀스 길이는 512로 고정하고 다른 종류의 시계열 데이터에 대한 유사 매칭 수행 시간을 측정하였다.

그림 5는 이와 같은 실험 결과를 나타낸다. 그림 5(a)는 질의 길이 변화에 따른 순차 검색과 제안한 방법의 유사 매칭 시간을 측정한 결과이며, 그림 5(b)는 다른 종류의 데이터를 이용하여 유사 매칭 시간을 측정한 결과이다. 먼저 그림 5(a)에서 알 수 있듯이, 제안한 인덱스 기반 서브시퀀스 매칭 방법은 순차 검색과 비교하여 4배 이상의 성능 개선 효과를 갖는다. 또한 그림 5(b)에서도 마찬가지로, 다른 종류의 데이터들의 경우에도 제안한 방법의 성능이 5배 이상 우수함을 볼 수 있다. 이는 실제 대용량 데이터에 적용을 하더라도 순차 검색 방법보다 제안한 인덱스 기반의 선형 추세 제거 서브시퀀스 매칭 방법의 성능이 더 뛰어날 것임을 알 수 있다.



(a) 질의 시퀀스 길이 변화에 따른 매칭 시간 측정 (b) 데이터 종류의 변화에 따른 매칭 시간 측정

그림 5 질의 시퀀스의 길이와 데이터 시퀀스의 종류에 따른 수행 시간 실험 결과

위와 같은 실험 결과를 통하여 제안한 선형 추세 제거 서브시퀀스 매칭은 실제 시계열 데이터의 선형 추세를 제거하여 유사 매칭을 수행하는 데 있어서 일반적인 순차 검색 방법보다 효과적임을 증명하였다.

5. 결 론

본 논문에서는 시계열 서브시퀀스 매칭 문제에 선형 추세 제거를 고려하는 문제를 제시하고, 이의 효율적인 해결책을 제안하였다. 본 논문의 공헌은 다음과 같다. 첫째, 선형 추세 제거 서브시퀀스 매칭의 순차 검색 방법을 제시하고, 이의 문제점을 분석하였다. 둘째, *LD*-윈도우의 개념을 제시하고, 이를 이용한 인덱스 기반 해결책을 제시하였다. 제안한 인덱스 기반 해결책은 인덱스 구성 알고리즘과 서브시퀀스 매칭 알고리즘으로 구성되어 있다. 셋째, 실험 결과를 통해 제안한 방법의 우수성을 입증하였다. 실제 시계열 데이터를 이용한 성능 평가 결과, 제안한 인덱스 기반 검색 방법이 순차 검색보다 수 배에서 수십 배 빠른 성능을 나타냈다. 본 논문에서 제안한 방법은 시계열 데이터에 나타나는 증가 혹은 감소 추세로 인해 알 수 없었던 데이터의 증감 폭의 패턴이 유사한 시계열을 찾아내는데 유용하게 사용될 수 있다.

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] Keogh, E., "A Decade of Progress in Indexing and Mining Large Time Series Databases," In *Proc. the 32nd Int'l Conf. on Very Large Data Bases*, Tutorial, Seoul, Korea, p.1268, Sept. 2006.

[4] 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 Trans. on Information and Systems*, vol. E84-D, no.1, pp.76-86, 2000.

[5] 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.

[6] Moon Y.-S., and Kim, J., "Efficient Moving Average Transform-Based Subsequence Matching Algorithms in Time-Series Databases," *Information Sciences*, vol.177, no.23, pp.5415-5431, Dec. 2007.

[7] Moon, Y.-S., and Kim, J., "Fast Normalization-Transformed Subsequence Matching in Time-Series Databases," *IEICE Trans. on Information and Systems*, vol.E90-D, no.12, pp.2007-2018, Dec. 2007.

[8] Hill, R. C., Griffiths, W. E., and Judge, G. G., Undergraduate Econometrics, John Wiley & Sons Inc., 2007.