시계열 데이터베이스에서 단일 색인을 사용한 정규화 변환 지원 서브시퀀스 매칭*

문양세○ 김진호
강원대학교 자연과학대학 컴퓨터과학과, 한국과학기술원 첨단정보기술연구센터
{ysmoon○, jhkim}@kangwon.ac.kr

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

Yang-Sae Moon○ Jinho Kim
Dept. of Computer Science, Kangwon Nat'l University and AITrc, KAIST

요 약

본 논문에서는 단일 색인을 사용하는 정규화 변환 지원 서브시퀀스 매칭 방법을 제안한다. 기존의 정규화 변환 지원 서브시퀀스 매칭 방법은 질의 시퀀스 길이가 커질수록 성능이 저하되고, 이를 해결하기 위하여 여러 개의 색인을 사용하는 방법을 취하였다. 본 논문에서는 하나의 색인을 사용하면서도 다양한 길이의 정규화 변환 지원 서브시퀀스 매칭을 수행하는 효율적인 방법을 제시한다. 이를 위하여, 본 논문에서는 정규화 변환의 정의를 확장하여 일반화 정규화 변환 개념을 제시한다. 또한, 이러한 일반화 정규화 변환 개념을 기존 서브시퀀스 매칭 방법들에 적용하는 방안에 대한 이론적 근거를 각각의 정리로서 제시하고 증명하였다. 그리고, 이들 방안을 구현하기 위한 색인 구성 알고리즘 및 서브시퀀스 매칭 알고리즘을 각각 제시하였다. 본 논문에서 제안한 정규화 변환 지원 서브시퀀스 매칭은 다른 변환을 지원하는 서브시퀀스 매칭으로 일반화 될 수 있는 우수한 연구결과라 사료된다.

1. 서 론

시계열 데이터는 각 시간별로 측정한 실수 값의 시퀀스로, 그 예로는 주식 데이터, 환율 데이터, 날씨 변동 데이터 등이 있다. 시계열 데이터베이스에 저장된 시계열 데이터를 **데이터 시퀀스**라 부른다. 그리고, 사용자에 의해 주어진 **질의 시퀀스**와 유사한 데이터 시퀀스를 검색하는 방법을 **유사 시퀀스 매칭**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**인 ε 이하이면, 두 시퀀스 X 와 Y 는 **유사**하다고 한다[1-4]. 그리고, 본 논문에서는 거리 함수 $D(X,Y)$ 로 유클리디안 거리 함수를 사용하며[1-4], $D(X,Y)$ 가 ε 이하이면 X 와 Y 는 **ε-매치**한다고 정의한다. 그리고, 유클리디안 거리 함수가 갖는 문제점을 보완하기 위하여 많은 변환 기법이 사용되었는데[3,6], 본 논문에서는 이중 정규화 변환을 지원하는 문제를 다룬다. 본 논문에서 사용하는 주요 표기와 이에 대한 정의 및 의미는 표 1과 같다.

표 1. 주요 표기법.

기호	정의/의미
$S[i:j]$	S 의 i 에서 j 번째 엔트리까지로 구성된 서브시퀀스
$\mu(S)$, $\sigma(S)$	S 에 포함된 모든 엔트리의 평균과 표준편차
$\overline{S}$	S 를 정규화 변환한 시퀀스 ($\overline{S}[i]=(S[i]-\mu(S))/\sigma(S)$)
$\overline{S}[i:j]$	정규화 변환된 시퀀스 $\overline{S}$ 의 i 에서 j 번째 엔트리까지로 구성된 서브시퀀스
$\overline{S[i:j]}$	서브시퀀스 $S[i:j]$ 를 정규화 변환한 시퀀스
s_i	시퀀스 S 의 i 번째 디스조인트 윈도우 ($=S[(i-1)\cdot\omega+1:i\cdot\omega]$, $i\geq 1$)
$\overline{s_i}$	정규화 변환된 시퀀스 $\overline{S}$ 의 i 번째 디스조인트 윈도우

본 논문에서는 정규화 변환을 서브시퀀스 매칭에 적용하는 문제를 다룬다. 즉, 질의 시퀀스와 데이터 서브시퀀스를 정규화 변환한 이후의 두 시퀀스가 ε-매치는지를 판단한다. 그리고, 이와 같이 정규화 변환한 이후에 두 시퀀스 간의 유클리디안 거리로 유사 시퀀스 매칭을 수행하는 방법을 본 논문에서는 **정규화변환 서브시퀀스 매칭**이라 정의한다. 즉, 정규화 변환 서브시퀀스 매칭이란 질의 시퀀스 Q 와 허용치 ε이 주어졌을 때, 데이터 시퀀스 S 의 서브시퀀스 중에서 $D(\overline{Q},\overline{S[i:j]})$ 가 허용치 ε 이하인 서브시퀀스 $S[i:j]$ 를 찾는 문제이다[3].

정규화 변환 서브시퀀스 매칭에 대해서는 Loh 등[3]에 의해 효과적인 알고리즘이 제안되었다. Loh 등이 제안한 방법은 비교 대상이 되는 두 시퀀스를 일정한 크기의 윈도우로 나누어 정규화 변환했을 때, 이들 변환된 윈도우의 거리가 ε'보다 작으면, 원래의 두 시퀀스를 정규화 변환한 거리가 ε 보다 작다는 정리를 사용하였다. 그리고, 이러한 한계 값은 ε'을 제시하였다. 또한, 기존 매칭 방법들과 마찬가지로 Loh 등은 데이터 시퀀스를 나눈 윈도우들을 다차원 색인에 저장하는 방법을 사용하였다[1,2,4]. 그런데, Loh 등의 방법은 질의 시퀀스가 지정된 윈도우 크기보다 커질 경우 성능이 저하되는 문제점이 있다. 따라서, Loh 등의 연구에서는 이러한 단점을 해결하기 위하여, 다양한 크기의 윈도우에 대해서 여러 개의 다차원 색인을 생성하는 색인 보간법을 사용하였다.

본 논문에서는 하나의 색인을 사용하면서도 정규화 변환 서브시퀀스 매칭을 효율적으로 수행하는 새로운 접근법을 제안한다. 우선, 서브시퀀스 $S[a:b]$ 에 대해, $S[a:b]$ 를 포함하는 $S[i:j]$ 의 평균과 표준편차를 사용하여 정규화 변환하는 **일반화 정규화 변환** 개념을 제시한다. 다음 표 2는 이와 같은 일반화 정규화 변환의 표기법을 나타낸다. 이러한 일반화 정규화 변환 개념은 정규화 변환된 $\overline{S[a:b]}$ 는 $S[a:b]$ 를 포함하는 $S[i:j]$ 의 평균과 표준편차를 사용하여 일반화 정규화 변환한 $\overline{S_{[i:j]}[a:b]}$ 와 유사할 것이라는 관찰에 기반한다. 다음으로, 이러한 일반화 정규화 변환의 개념을 사용하면, 기존의 서브시퀀스 매칭[2,4]에서 사용되었던 충분조건을 그대로 활용할 수 있음을 보이고, 이에 대한 이론적 근거를 제시한다.

표 2. 일반화 정규화 변환의 표기법.

기호	정의/의미
$\overline{S_{[i:j]}[a]}$	시퀀스 S 의 엔트리 $S[a]$ 를 $\mu(S[i:j])$ 과 $\sigma(S[i:j])$ 로 정규화 변환한 엔트리 ($=(S[a]-\mu(S[i:j]))/\sigma(S[i:j])$)
$\overline{S_{[i:j]}[a:b]}$	서브시퀀스 $S[a:b]$ 를 $\mu(S[i:j])$ 과 $\sigma(S[i:j])$ 으로 정규화 변환한 시퀀스($i\leq a<b\leq j$)

다음으로, 일반화 정규화 변환 개념을 서브시퀀스 매칭의 기존 연구인 Faloutsos 등의 연구[2](간략히, ***FRM***이라 한다) 및 DualMatch[4]에 적용하는 방법을 제안한다. 즉, FRM 및 DualMatch에서 취한 윈도우 구성법에 일반화 정규화 변환 개념을 적용할 수 있 음을 보인다. 그리고, 일반화 윈도우를 적용하는 방법이 서브시퀀스 매칭을 정확하게 수행함을 정리로 제시하고 증명한다. 또한, 이를 구현하기 위한 색인 구성 알고리즘 및 서브시퀀스 매칭 알고리즘을 제안한다.

2. 관련 연구

우선, Agrawal 등[1]의 전체 매칭을 색인 구성 및 유사 시퀀스 매칭 알고리즘으로 구분하여 설명한다. 색인 구성 알고리즘에서는 길이 n 인 데이터 시퀀스를 f-차원($f\leq n$)의 점으로 변환하여 f-차원의 R*-트리[5]에 저장

* 본 연구는 첨단정보기술연구센터를 통해 한국과학재단의 지원을 받았음

한다. 다음으로, 유사 시퀀스 매칭 알고리즘에서는 질의 시퀀스를 f-차원 점으로 변환하고, 변환한 점과 허용치 ε으로 R*-트리를 검색하여, ε-매치하는 모든 점들을 찾아 **후보집합**을 구한다. 이렇게 후보집합을 구하면 **착오기각**(유사 시퀀스이나 착오로 인해 기각되는 데이터 시퀀스)은 발생하지 않지만, **착오해답**(후보이나 실제로는 질의 시퀀스와 ε-매치하지 않는 데이터 시퀀스)이 발생할 수 있다. 따라서, 각 후보 시퀀스들에 대해서 실제 데이터 시퀀스를 액세스하고 질의 시퀀스와의 거리를 조사하여 착오해답을 제거하는 **후처리 과정**을 수행한다[1].

Faloutsos 등[2]은 전체 매칭을 일반화하여 서브시퀀스 매칭을 처음 소개하였다. FRM에서는 데이터 시퀀스를 슬라이딩 윈도우로 나누고 질의 시퀀스를 디스조인트 윈도우로 나누는 방법을 사용한다. FRM의 색인 구성 알고리즘에서는 데이터 시퀀스를 나눈 슬라이딩 윈도우로 f-차원의 점으로 변환하고, 변환된 여러 점을 포함하는 MBR(minimum bounding rectangle)을 구성한 후, 이들 MBR을 다차원 색인인 R*-트리에 저장한다. 그리고, FRM에서는 다음 보조정리 1에 기반하여 서브시퀀스 매칭 알고리즘을 제안하였다.

보조정리 1[2]: 데이터 시퀀스 S를 크기 ω인 슬라이딩 윈도우로 나누고, 질의 시퀀스 Q를 같은 크기의 디스조인트 윈도우로 나누었을 때, 길이 $Len(Q)$ 인 S의 서브시퀀스 $S[i:j]$와 Q가 ε-매치하면, Q에 포함된 적어도 하나 이상의 윈도우 q_k ($1 \le k \le p$)와 $S[i:j]$에 포함된 윈도우 $S[i+(k-1)\cdot\omega : i+k\cdot\omega-1]$이 $\varepsilon/\sqrt{p}$-매치한다($p=\lfloor Len(Q)/\omega \rfloor$).

보조정리 1에 따르면, q_k가 $S[i+(k-1)\cdot\omega : i+k\cdot\omega-1]$와 $\varepsilon/\sqrt{p}$-매치할 때, 서브시퀀스 $S[i:j]$로 후보집합을 구성하면, 착오기각 없이 모든 유사 서브시퀀스를 구할 수 있다.

FRM의 성능을 개선한 DualMatch[5]에서는 데이터 시퀀스를 디스조인트 윈도우로 나누고, 질의 시퀀스를 슬라이딩 윈도우로 나누는 방법을 사용한다. 이는 FRM에서 다차원 색인에 점이 아닌 MBR을 저장함으로써 발생하는 착오해답 증가의 문제를 해결하기 위함이다. DualMatch에서는 데이터 시퀀스를 나눈 디스조인트 윈도우를 f-차원의 점으로 변환하여 R*-트리에 저장한다. 그리고, 다음 보조정리 2에 기반하여 착오기각없이 서브시퀀스 매칭을 수행한다.

보조정리 2[5]: 데이터 시퀀스 S를 크기 ω인 디스조인트 윈도우로 나누고, 질의 시퀀스 Q를 같은 크기의 슬라이딩 윈도우로 나누었을 때, 길이 $Len(Q)$ 인 S의 서브시퀀스 $S[i:j]$와 Q가 ε-매치하면, $S[i:j]$에 포함된 적어도 하나 이상의 윈도우 $S[i+k:i+k+\omega-1]$과 Q에 포함된 윈도우 $Q[k:k+\omega-1]$이 $\varepsilon/\sqrt{p'}$-매치한다($p'=\lfloor (Len(Q)+1)/\omega \rfloor - 1$).

보조정리 2에 따르면, $Q[k:k+\omega-1]$과 $S[i+k:i+k+\omega-1]$이 $\varepsilon/\sqrt{p'}$-매치할 때, 서브시퀀스 $S[i:j]$로 후보집합을 구성하면 착오기각 없이 모든 유사 서브시퀀스를 구할 수 있다.

그런데, 정규화 변환 서브시퀀스 매칭을 위하여 앞서의 유클리디안 거리 기반 서브시퀀스 매칭인 FRM과 DualMatch를 그대로 사용할 수 없다. 그 이유는 유클리디안 거리 기반 서브시퀀스 매칭에서 성립하던 보조정리 1과 2가 정규화 변환에 그대로 적용되지 않기 때문이다. Loh 등[3]은 이러한 문제점을 해결하는 새로운 정규화 변환 서브시퀀스 매칭 방법을 제안하였다. Loh 등은 다음 조건식 (1)이 성립함을 증명하고, 보조정리 1 및 2 대신에 다음 식 (1)를 사용하였다.

$$D(\overline{S},\overline{Q}) \le \varepsilon \Rightarrow D(\overline{S[i:j]},\overline{Q[i:j]}) \le \varepsilon' \qquad (1)$$

식 (1)에서 ε'은 $\sqrt{2\omega - 2\sqrt{\omega^2-\omega\cdot\varepsilon^2\cdot\left(\sigma^2(Q)/\sigma^2(Q[i:j])\right)}}$ 이다. Loh 등은 식 (1)이 성립함을 증명하고, 정규화 변환된 데이터 윈도우와 질의 윈도우가 ε'-매치할 때, 해당 데이터 윈도우를 포함하는 서브시퀀스를 후보로 삼는 방법을 사용하였다.

3. 연구 동기

Loh 등의 매칭 방법은 질의 시퀀스 길이가 색인 구성에 사용된 윈도우 크기와 다를 경우 상당한 성능 저하가 발생한다[3]. 즉, 색인이 구성되지 않은 길이의 질의 시퀀스의 경우, 탐색 범위가 커져 색인 페이지 액세스가 많아질 뿐 아니라 후보 개수 증가로 인한 후처리 과정의 데이터 페이지 액세스가 많아지게 된다. 이러한 문제점을 해결하기 위하여, Loh 등은 여러 개의 인덱스를 사용하는 색인 보간법을 제안하였다. 색인 보간법이란 여러 크기의 윈도우에 대해서 다차원 색인을 구성하고, 질의 시퀀스 길이에 따라 적절한 다차원 색인을 선택하여 서브시퀀스 매칭을 수행하는 방법이다. 그러나, 이러한 색인 보간법을 사용할 경우, 색인 저장 공간의 오버헤드와 여러 색인에 대한 관리 오버헤드가 발생하는 문제점이 있다.

본 연구의 동기는 기존 서브시퀀스 매칭의 보조정리 1 및 2를 정규화 변환 서브시퀀스 매칭에 적용하자는 데 있다. 그런데, 보조정리 1 및 2를 사용하기 위해서는 각각의 윈도우에 대해서 다양한 길이 및 다양한 위치의 서브시퀀스로 일반화 정규화 변환하는 작업이 필요하다. 즉, 시퀀스 S를 나눈 윈도우 $S[a:b]$에 대해서, $S[a:b]$를 포함하는 모든 $S[i:j]$를 사용하여 일반화 정규화 변환하여 관리하는 과정이 필요하다. 이러한 문제는 시계열 데이터의 경우, 이웃한 엔트리의 값의 변화가 크지 않다는 관찰에 기반하여 해결할 수 있다. 즉, 시퀀스 S를 나눈 윈도우 하나를 $S[a:b]$ 이라 하고, $S[a:b]$를 포함하는 서브시퀀스를 $S[i:j]$ 라 했을 때, $Len(S[i:j])$ 와 $Len(S[a:b])$ 의 차이가 크지 않다면, $S[a:b]$를 정규화 변환한 시퀀스 $\overline{S[a:b]}$ 와 $S[a:b]$ 를 $S[i:j]$ 로 일반화 정규화 변환한 시퀀스 $\overline{S[i:j][a:b]}$ 는 유사하게 된다. 결과적으로, 윈도우 $S[a:b]$를 일반화 정규화 변환한 여러 개의 $\overline{S[i:j][a:b]}$는 유사한 값을 가질 것이고, 본 논문에서는 이러한 성질을 사용하는 정규화 변환 서브시퀀스 매칭을 제안한다.

4. 단일 색인을 사용한 정규화 변환 서브시퀀스 매칭

4.1 개념

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

보조정리 3: 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$에 윈도우 $S[a:b]$가 포함되어 있고, $S[i:j]$와 비교 대상이 되는 질의 시퀀스를 Q라 하면, 다음 식 (2)의 충분조건이 성립한다.

$$D(\overline{S[i:j]},\overline{Q}) \le \varepsilon \Rightarrow D(\overline{S_{[i:j]}[a:b]},\overline{Q[a-i+1:b-j+1]}) \le \varepsilon \qquad (2)$$

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

보조정리 4: 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$에 같은 크기인 p개의 윈도우 $S[a_0:a_1-1]$, $S[a_1:a_2-1]$, ..., $S[a_{p-1}:a_p-1]$가 포함되어 있다고 하고, $S[i:j]$와 비교 대상이 되는 질의 시퀀스를 Q라 하면, 다음 식 (3)의 충분조건이 성립한다.

$$D(\overline{S[i:j]},\overline{Q}) \le \varepsilon \Rightarrow \bigvee_{k=1}^{p} D(\overline{S_{[i:j][a_{k-1}:a_k-1]}},\overline{Q[a_{k-1}-i+1:a_k-i]}) \le \varepsilon/\sqrt{p} \qquad (3)$$

4.2 FRM을 사용한 해결 방법

본 절에서는 FRM의 윈도우 구성법에 일반화 정규화 변환을 적용하여 정규화 변환 서브시퀀스 매칭을 수행하는 방법을 제안한다(간략히 *NFRM*(*Normalization FRM*)이라 한다). 제안하는 NFRM은 다음 정리 1에 기반하여 정규화 변환 서브시퀀스 매칭을 수행한다.

정리 1: 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$를 변환한 $\overline{S[i:j]}$와 질의 시퀀스 Q를 정규화 $\overline{Q}$가 ε-매치한다면, 적어도 하나 이상의 $\overline{S[i:j]}$에 포함된 크기 ω인 윈도우 $\overline{S_{[i:j][i+(k-1)\cdot\omega : i+k\cdot\omega-1]}}(1 \le k \le p)$ 와 같은 크기인 $\overline{Q}$의 k 번째 디스조인트 윈도우 $\overline{q_k}$가 $\varepsilon/\sqrt{p}$-매치한다($p = \lfloor Len(Q)/\omega \rfloor$).

결국, NFRM에서는 정리 1에 따라서, $\overline{S_{[i:j][i+(k-1)\cdot\omega : i+k\cdot\omega-1]}}$와 $\overline{q_k}$가 $\varepsilon/\sqrt{p}$-매치할 때, $\overline{S[i:j]}$를 $\overline{Q}$ 후보 서브시퀀스로 삼으면 착오기각이 발생하지 않는다.

정리 1에 기반한 NFRM의 색인 구성 알고리즘은 그림 1과 같다. 또한, 다차원 색인이 구성된 후에는 그림 2의 서브시퀀스 매칭 알고리즘을 사용하여 정규화 변환 서브시퀀스 매칭을 수행한다.

Procedure *NFRM-PointIndex*(Data Sequence S, Window size ω)
(1) Divide S into sliding windows of length ω;
(2) **for** each sliding window $S[a:b]$ **do**
(3) for each query length q_{len} **do**
(4) for each subsequence $S[i:j]$ ($i=a-k\cdot\omega$, $j=i+q_{len}-1$) **do**
(5) Construct a transformed window $\overline{S_{[i:j]}[a:b]}$;
(6) Transform the window to an *f-point*;
(7) Make a record <*f-point*, *offset*=*i*, q_{len} >, and store it to the index;
(8) **endfor**
(9) **endfor**
(10) **endfor**

그림 1. NFRM에서 점을 저장하는 색인 구성 알고리즘.

Procedure *NFRM-PointMatching* (Query Sequence Q, Window size ω)
(1) Make $\overline{Q}$ from Q by using the normalization transform;
(2) Divide $\overline{Q}$ into disjoint windows $\overline{q_i}(1\le i\le p)$ of length ω;
(3) **for** each disjoint window $\overline{q_i}$ **do**
(4) Transform the window to an *f*-dimensional point;
(5) Construct a range query using the point and $\varepsilon/\sqrt{p}$;
(6) Find the records of the form <*f-point*, *offset*, q_{len}> from the index;
(7) Include in the candidate set $S[offset:offset+q_{len}-1]$;
(8) **endfor**
(9) Do the post-processing step;

그림 2. NFRM에서 점 색인을 사용한 서브시퀀스 매칭 알고리즘.

그런데, NFRM에서 점을 저장하는 색인 구성 방법을 사용할 경우, 색인에 저장해야 하는 점의 개수가 너무 많아지는 문제점이 있다. 이러한 문제점을 해결하기 위해서, 실제로 FRM에서는 여러 개의 점을 포함하는 MBR을 구성하는 방법을 사용하였다[2]. 이에 따라, FRM과 같이 MBR을 사용하여 여러 개의 점을 포함하는 형태로 색인 구성 알고리즘을 보완하면 다음 그림 3 및 4와 같다. 그리고, 이들 알고리즘은 향후 여러 개의 슬라이딩 윈도우들이 매핑된 여러 개의 MBR을 하나의 MBR에 포함시키는 것으로 확장될 수 있다.

Procedure *NFRM-PointIndex*(Data Sequence S, Window size ω)
(1) Divide S into sliding windows of length ω;
(2) **for** each sliding window $S[a:b]$ **do**
(3) Construct a new record <*f-mbr* = *empty*, *offset* = *a*>;
(4) for each query length q_{len} **do**
(5) for each subsequence $S[i:j]$ ($i=a-k\cdot\omega$, $j=i+q_{len}-1$) **do**
(6) Construct a transformed window $\overline{S_{[i:j]}[a:b]}$;
(7) Transform the window to an *f-point*;
(8) Include in the *f-mbr* the transformed point *f-point*;
(9) **endfor**
(10) Store the record <*f-mbr*, *offset* > into the index;
(11) **endfor**
(12) **endfor**

그림 3. NFRM에서 MBR을 저장하는 색인 구성 알고리즘.

Procedure *NFRM-PointMatching* (Query Sequence Q, Window size ω)
(1) Make $\overline{Q}$ from Q by using the normalization transform;
(2) Divide $\overline{Q}$ into disjoint windows $\overline{q_i}(1\le i\le p)$ of length ω;
(3) **for** each disjoint window $\overline{q_i}$ **do**
(4) Transform the window to an *f*-dimensional point;
(5) Construct a range query using the point and $\varepsilon/\sqrt{p}$;
(6) Search the *index* and find the records of the form <*f-mbr*, *offset*>;
(7) Include in the candidate set $S[i:j]$ ($i=offset-k\cdot\omega$, $j=i+Len(Q)-1$);
(8) **endfor**
(9) Do the post-processing step;

그림 4. NFRM에서 MBR 색인을 사용한 서브시퀀스 매칭 알고리즘.

4.3 DualMatch를 사용한 해결 방법

본 절에서는 DualMatch의 윈도우 구성법에 일반화 정규화 변환을 적용하여 정규화 변환 서브시퀀스 매칭을 수행하는 방법을 제안한다(간략히 *NDM*(*Normalization DualMatch*)이라 한다). NDM은 다음 정리 2에 기반하여 정규화 변환 서브시퀀스 매칭을 수행한다.

정리 2: 데이터 시퀀스 S의 서브시퀀스 $S[i:j]$를 정규화 변환한 $\overline{S[i:j]}$와 질의 시퀀스 Q를 정규화 변환한 $\overline{Q}$가 ε-매치한다면, 적어도 하나 이상의 $\overline{S[i:j]}$에 포함된 크기 ω인 윈도우 $\overline{S_{[i:j]}[a:b]}$ ($a=\lceil(i-1)/\omega\rceil\cdot k\cdot\omega+1$, $b=a+\omega-1$, $1\le k\le p'$)와 같은 크기인 $\overline{Q}$의 $a-i+1$ 번째 슬라이딩 윈도우인 $\overline{Q}[a-i+1:a-i+\omega]$가 $\varepsilon/\sqrt{p'}$-매치한다($p'=\lfloor(Len(Q)+1)/\omega\rfloor-1$).

결국, NDM에서는 정리 2에 의해서, NDM은 $\overline{Q}$의 c 번째 슬라이딩 윈도우인 $\overline{Q}[c:c+\omega-1]$과 $\overline{S_{[c-a+1:c-a+\omega]}[a:a+\omega-1]}$가 $\varepsilon/\sqrt{p'}$-매치할 때, 시작 위치를 i ($=c-a+1$)로 하는 서브시퀀스 $\overline{S[i:j]}$를 $\overline{Q}$와 ε-매치하는 후보 서브시퀀스로 삼으면 착오기각이 발생하지 않는다.

정리 2에 기반한 NDM의 색인 구성 및 서브시퀀스 매칭 알고리즘은 NFRM의 색인 구성 및 서브시퀀스 매칭 알고리즘과 유사하다. 지면 관계상, 본 논문에서는 NDM의 알고리즘 기술을 생략한다.

5. 결론

본 논문에서는 하나의 색인을 사용하여 정규화 변환을 지원하는 서브시퀀스 매칭을 효율적으로 수행하는 방법을 제안하였다. 기존의 정규화 변환 지원 서브시퀀스 매칭 방법은 질의 시퀀스 길이가 윈도우 크기보다 클 경우 검색 범위가 커져 성능이 저하되는 문제점이 있으며, 이를 해결하기 위하여 여러 개의 색인을 사용하는 방법을 취하였다. 반면에, 제안한 정규화 변환 지원 서브시퀀스 매칭 방법은 일반화 정규화 변환 개념을 사용하여 하나의 색인을 사용해서도 다양한 길이의 질의 시퀀스에 대해 효율적인 서브시퀀스 매칭을 수행할 수 있다. 본 논문에서는 정규화 변환을 확장하여 일반화 정규화 변환 개념을 정형적으로 정의하고, 이를 기존 서브시퀀스 매칭 방법인 FRM 및 DualMatch에 적용하는 방안에 대한 이론적 근거를 각각의 정리로서 제시하였다. 또한, 이들 방안을 구현하기 위해서 점을 직접 저장하는 색인 구성 알고리즘 및 서브시퀀스 매칭 알고리즘을 FRM 및 DualMatch에 대해서 각각 제시하였다.

본 논문에서 제안한 정규화 변환 지원 서브시퀀스 매칭은 다른 변환을 지원하는 서브시퀀스 매칭의 일반화 될 수 있다. 따라서, 제안한 방법은 정규화 변환을 포함하는 많은 다른 종류의 변환을 지원하는 서브시퀀스 매칭에 폭넓게 적용될 수 있을 것으로 사료된다. 본 논문의 향후 연구로는 실험을 통하여 제시한 방법의 우수성을 입증하는 것이다.

참고문헌

[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] 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, pp. 5-28, 2004.

[4] 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*, Heidelberg, Germany, pp. 263-272, April 2001.

[5] 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.

[6] 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*, Orlando, Florida, pp. 201-208, Feb. 1998.