Journal of Korean Institute of Intelligent Systems, Vol. 22, No. 2, April 2012, pp. 181-186
http://dx.doi.org/10.5391/JKIIS.2012.22.2.181

# 시퀀스 데이터웨어하우스에서 이산푸리에변환과 비트맵을 이용한 시퀀스 스트림 색인 기법

# Sequence Stream Indexing Method using DFT and Bitmap in Sequence Data Warehouse

손동원.홍동권†

Dong-won Son, Dong-Kweon Hong†

계명대학교 컴퓨터공학과

**요 약**

최근 시간적으로 변화된 데이터에서 유사한 값의 움직임 즉 유사 패턴을 검색하는 연구가 활발히 진행되고 있다. 시간적으로 변화된 데이터는 시계열 데이터 (time series data) 또는 시퀀스 데이터(sequence data)로 분류되며 기존의 스칼라 값을 가지는 데이터와는 매우 다른 의미를 가진다. 본 논문에서 유사 시퀀스 검색은 시퀀스 데이터웨어하우스에서 값의 변화가 유사한 형태를 가지는 시퀀스들을 검색한다. 유사 시퀀스를 검색하기 위하여 본 논문에서는 먼저 시퀀스 원시 데이터에 이산 푸리에 변환(DFT, Discrete Fourier Transform)을 적용하여 데이터를 변환한다. 변환된 데이터는 그 특성으로 인하여 유사 패턴의 검색에 적합하며 또 유사도를 비교할 때 일부분만 사용되므로 색인에 사용되는 속성의 개수를 줄이는 장점이 있다. 또 데이터웨어하우스 환경이므로 더 좋은 성능을 보일 수 있는 비트맵 색인 기법을 적용하였다. 시퀀스 데이터의 효율적인 검색을 위하여 영역 지정 검색 방법을 제안하고 효율적인 실행을 위한 비트맵을 활용한 다양한 조합의 색인을 생성하고, 질의 최적화기의 연산 비용을 비교하면서 효율적인 검색 연산을 위한 최저 비용의 색인을 선택하는 기법을 연구하였다.

**키워드 :** 시퀀스 데이터, 데이터 스트림, 유사 검색, 이산 푸리에 변환, 비트맵 색인

**Abstract**

Recently there has been many active researches on searching similar sequences from data generated with the passage of time. Those data are classified as time series data or sequence data and have different semantics from scalar data of traditional databases. In this paper similar sequence search retrieves sequences that have a similar trend of value changes. At first we have transformed the original sequences by applying DFT. The converted data are more suitable for trend analysis and they require less number of attributes for sequence comparisons. In addition we have developed a region-based query and we applied bitmap indexes which could show better performance in data warehouse. We have built bitmap indexes with varying number of attributes and we have found the least cost query plans for efficient similar sequence searches.

**Key Words :** sequence data, data stream, similarity search, DFT, bitmap index

## 1. 서 론

주식 값의 변화, 노래 멜로디의 변화, 기후 변화 등 시간적으로 변화된 데이터에서 유사한 값의 움직임 즉 유사 패턴을 검색하는 연구의 중요성이 점점 더 강조되면서 관련 기술의 연구가 활발히 진행되고 있다 [1, 2, 3]. 시간적으로 변화된 데이터는 시계열 데이터 (time series data) 또는 시퀀스 데이터 (sequence data)로 분류되며 기존의 스칼라 값을 가지는 데이터와는 매우 다른 의미를 가진다. 시퀀스 데이터에서의 데이터 검색 방법은 저장된 데이터와 질의 데이터를 비교하여 비슷한 패턴을 검색하는 "유사검색 (similarity search)"으로 변화되었다. 이는 기존 스칼라 값의 비교에서 전체 또는 부분적인 값이 일치하는 것을 검색하는 것과는 전혀 다른 새로운 연산을 요구하게 되었다. 시퀀스 연산에서는 비록 시퀀스를 구성하는 구성 성분 각각의 값이 달라도 전체적인 시퀀스의 흐름이 비슷할 경우 우리는 2개의 시퀀스를 유사하다고 한다.

대용량 시퀀스에서 질의 시퀀스와 유사한 것을 찾기 위해서 각각의 시퀀스를 직접 비교하는 것은 너무 많은 시간이 걸려 대부분의 응용에서는 직접 비교보다 훨씬 더 빠른 방법을 요구하고 있다. 따라서 본 논문에서는 검색 시퀀스와 데이터베이스에 저장된 시퀀스의 길이가 같다는 가정에서 비슷한 정도, 즉 유사도가 ε 범위에

접수일자: 2012년 1월 18일
심사(수정)일자: 2012년 3월 27일
게재확정일자 : 2012년 3월 29일
†교신저자
본 연구는 2010년도 계명대학교 비사연구기금으로 이루어졌음.

존재하는 모든 시퀀스를 효율적으로 검색하기 위한 색인 기법을 연구한다. 유사 범위를 나타내는 파라미터 ε는 응용 도메인에 따라 적합한 값을 설정하게 되며 사용자, 응용 분야의 전문가 또는 응용 프로그램에서 자동으로 결정하게 된다.

본 논문에서 시퀀스가 저장되는 곳은 시퀀스 데이터가 입출력이 빈번하지 않으며 지속적으로 시퀀스가 추가되는 시퀀스의 특성에 따라 시퀀스에 적용되는 연산이 OLTP (On-Line Transaction Processing)가 아니라 OLAP(On-Line Analytic Processing)와 데이터 마이닝(data mining)이 적용되는 데이터웨어하우스 (data warehouse)이다. 데이터웨어하우스에서는 그 특성상 효율적인 연산을 위한 전처리 (pre-processing) 시간이 소비되어도 전체 성능에 영향을 미치지 않으므로 본 논문에서의 전처리 과정은 시간 도메인으로 구성된 시퀀스 원시 데이터에 DFT (Discrete Fourier Transform)를 적용하여 주파수 도메인 데이터로 변경한다. 변경된 데이터는 DFT의 특성으로 인하여 유사 패턴의 검색에 적합하며 또 유사도를 비교할 때 일부분만 사용되므로 색인에 사용되는 속성의 개수를 줄이는 장점이 있다. 뿐만 아니라 데이터웨어하우스는 기존 데이터의 변경이 거의 발생하지 않고, 적용되는 연산이 저장된 데이터의 많은 부분을 활용하는 특성으로 인하여 기존 데이터베이스와는 다른 방식의 색인 기법이 적용된다. 본 논문에서의 시퀀스 검색을 위한 과정은 대용량의 시퀀스 데이터웨어하우스에서 검색 시퀀스와 유사한 패턴을 가지는 즉 유사도 ε 범위 내에 있는 모든 검색 시퀀스를 찾기 위한 방법으로 원천 데이터를 시퀀스 데이터로 표현한 후 원천 데이터를 유사 검색에 적합하게 DFT를 사용하여 변환하였다. 그 다음 변환된 데이터를 효율적으로 검색하기 위한 인덱스를 구축하였고 마지막으로 제안된 방법의 성능 평가를 위한 실험 시스템을 구축하고 그 성능을 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개하고 3장에서는 유사검색 과정의 방법, 4장에서는 DFT를 이용한 유사검색 기능 평가, 5장에서는 DFT를 이용한 유사검색 성능 평가를 설명하고 마지막 6장에서는 결론을 제시한다.

## 2. 관련 연구

시퀀스 연구 분야에서 각각의 시퀀스는 n 포인트의 값으로 표현된다. 유사한 패턴을 가진 시퀀스를 찾는 전형적인 방법은 먼저 시퀀스에서 k개의 특징값을 추출해내고 (n >> k), 그 값을 k 차원의 값으로 맵핑한 후 다차원 인덱스를 사용하여 그 값을 검색하는 방법을 사용한다 [1]. 시퀀스에서 n개의 특징값을 추출하는 대표적인 방법으로는 시간 도메인의 시퀀스를 다른 도메인의 데이터로 변환하여 변환된 데이터의 일부분을 사용하는 기법이 있는데 그 대표적인 방법으로는 DFT를 사용하는 방법이 있다. n 포인트 DFT는 다음과 같이 요약할 수 있다. n 포인트 신호가 다음과 같을 때

$$\vec{x} = [x_t], t = 0,..,n-1$$

그 신호의 DFT 변환은 다음의 공식으로 계산된다.

$$X_f = 1/\sqrt{n}\sum_{t=0}^{n-1} x_t \exp(-j2\pi ft/n)\ f = 0,1,..,n-1$$
$$(j = \sqrt{-1})$$

DFT를 사용하는 방법은 Parseval의 이론 (Parselval's theorem)[1]에 의해 DFT는 시간 도메인과 주파수 도메인 사이의 유클리드 거리 (Euclidean distance)를 보존하므로 유사 시퀀스 검색에 적절한 성질을 가지고 있다. 하지만 어떤 변환 방법을 사용하던지 변환된 값에서 n개 이상의 속성 값을 그대로 사용하는 경우 다차원 문제점 (Dimensionality curse)에 의해 n개의 속성을 모두 사용하는 다차원 인덱스는 매우 비효율적이다 [4]. 따라서 n 차원을 사용하는 고차원 인덱스는 일반적으로 매우 비효율적이므로 n차원의 DFT 변환 계수에서 k개의 계수만 사용하는 (n >> k) 저차원 (low dimensionality)으로 변환한 후 R-tree 와 같은 다차원 인덱스를 사용하여 유사 검색을 지원한다. 하지만 비록 k값이 n에 비해 작은 값이지만 k (k > 3)차원을 효과적으로 지원하는 것은 여전히 어려운 일이다 [4]. 시퀀스 데이터베이스 연구 분야에서는 시퀀스의 변환 기법 외에도 시퀀스 유사 검색을 위한 새로운 모델의 제안과 [5, 6], 불완전한 시퀀스를 처리하기 위한 연구 기법 [7], 그리고 서브 시퀀스에 대한 다양한 연구가 활발히 진행되고 있다 [8, 9, 10]. 하지만 이런 연구들도 여전히 현재 대부분의 데이터를 보유하고 있는 상용 DBMS에 직접 적용하는 방안을 제시하지 못하고 있다.

## 3. 데이터웨어하우스의 시퀀스 유사 검색

본 논문에서의 시퀀스 검색은 대부분의 경우 원시 데이터에 직접 적용하기는 힘들다. 따라서 본 연구에서는 운용 데이터베이스의 데이터를 직접 사용하는 방법이 아니라 전처리 과정을 거친 후 데이터웨어하우스에 저장된 데이터를 사용하는 방식을 적용한다.

### 3.1 데이터웨어하우스에 데이터 표현 방법

먼저 데이터베이스 데이터를 데이터웨어하우스에 저장하기 위하여 필요한 데이터를 추출하였다. 추출된 데이터를 ID와 날짜로 정렬한 원시 데이터가 다음의 [표 1] 형식을 가진다고 가정할 때 다음 테이블의 내용을 정리하여 검색하기 쉽게 변환하였다.

표 1. 정렬된 원시 데이터
Table 1. Sorted original data

| ID | when | what | value |
|---|---|---|---|
| 1 | 2010.10.09 | 검사A | 160 |
| 1 | 2011.3.12 | 검사A | 150 |
| 1 | 2011.4.15 | 검사A | 140 |

먼저 [표 1]의 데이터를 데이터웨어하우스에 저장할 때 어떻게 표현할 것인지에 대한 연구를 진행 하였다. 관계형에서 원시 데이터를 활용하여 쉽게 표현하는 방법은

[표 1]과 같이 순수 관계형으로 표시하는 방법과, [표 2]와 같이 객체-관계형의 컬렉션 자료형으로 표현하는 2가지 방법이 있다. 2가지 방법은 장단점이 있으나 불규칙적인 시간 간격의 조정과 새로운 데이터의 삽입을 고려할 때 본 연구에서는 [표 1]의 방법을 사용하였다.

표 2. 시퀀스를 객체 관계형 테이블에 표현방법
Table 2. Representation of sequence in OR table

| ID | what | values sequence |
|---|---|---|
| 1 | 검사A | (80, 50, ....) |
| 2 | 검사A | (90, 80, ...) |

### 3.2 불규칙적인 시간 간격 조정

데이터웨어하우스에 저장되는 [표 1]의 데이터 형태는 일반적으로 시간 간격이 균등하지 않다. 즉 같은 시간 간격의 관점에서 데이터를 보면 이는 불완전한 데이터 (missing data)로 간주된다. 따라서 [표 1]의 데이터를 좀 더 의미 있는 형태의 시간 간격으로 데이터를 재정리 하여 표현하는 과정이 필요하다. 불완전한 데이터가 존재하는 시간 구간에서 데이터를 처리 하는 방법으로 다음의 3가지 방법을 고려한다. 3가지 방법은 데이터 도메인에 따라 다른 특성을 가지고 있다. 본 논문에서 다루는 개인의 정보 특성상 부정 오류 (false negative)보다는 긍정 오류 (false positive)가 더 의미를 가지므로 3)번의 방식을 채택한다.

1) 평균 추정값 (Average Estimation) - 양 끝 값의 평균으로 새로운 데이터를 생성하여 채운다.

2) 가장 최근값 (Last value Estimation) - 불완전 데이터가 존재하는 시간 구간의 가장 마지막 값으로 새로운 데이터를 생성하여 채운다.

3) 가장 이전값 (First value Estimation) - 불완전한 데이터가 존재하는 시간 구간의 처음 값으로 새로운 데이터를 생성하여 채운다.

### 3.3 데이터의 시퀀스 변환

[표 1]의 데이터를 3.2절의 방식에 의해 불완전한 데이터를 처리한 후 다음 단계는 [표 1]의 데이터를 검색 가능한 형태로 변환하는 과정을 수행 하였다. 본 연구에서는 [표 1]의 시퀀스를 DFT를 이용하여 변환한 후 DFT 계수의 초기 n개만 사용하는 방법을 적용한다. (n=3인 경우 실수와 허수 부분으로 구성되어 있으므로 6개의 값이 필요하다) 따라서 최초의 원시 데이터에서 다음의 [표 3] 테이블을 생성한다.

표 3. DFT를 사용하여 변환한 DFT_TABLE (n=3)
Table 3. DFT transformed DFT_TABLE (n=3)

| ID | what | DFT value of sequence | | | | | |
|---|---|---|---|---|---|---|---|
| | | n1 | n2 | n3 | n4 | n5 | n6 |
| 1 | 검사A | | | | | | |
| 2 | 검사A | | | | | | |

[표 3]의 검색 데이터와 유사한 시퀀스를 검색하는 내용도 따라서 DFT를 통하여 [표 3]의 데이터를 (v1, v2, v3, v4, v5, v6)의 값을 가지는 검색 시퀀스로 변환한다. 그 다음 검색 시퀀스와 가장 유사한 시퀀스를 [표 3]에서 검색한다.

### 3.4 영역 지정 방식의 유사 시퀀스 검색 방법

검색 데이터와 가장 유사한 것은 유클리드 거리가 가장 가까운 것 (nearest-neighbor)이므로 가장 유사한 것을 찾는 방법을 SQL의 상관 부질의 (Correlated subquery)를 사용하여 나타내면 다음의 [그림 1]과 같다. 이 경우 가장 가까운 것을 찾기 위하여 각각의 레코드에 대하여 테이블 전체를 비교해야 하는 매우 비효율적인 면이 있다. [그림 1]의 방식과 같이 인덱스를 사용하지 않는 경우 DFT_TABLE에 저장된 모든 데이터를 전부 스캔하기 위하여 디스크 IO가 많아지는 비효율성이 있다. 본 연구에서는 이 문제점을 해결하기 위한 방법으로 [표 3]의 컬럼 n1 ~ n6에 대한 인덱스를 적용하는 방식을 사용한다. 인덱스를 사용하기 위해서는 [그림 1]의 SQL 구문과 같이 각 컬럼의 값을 직접 비교하는 방식은 적합하지 않다. 대신 어떤 범위 내에 존재하는 레코드를 구하고, 그 다음 후처리 방식으로 최근접 레코드를 찾는 방식을 사용한다. 따라서 [그림 1]의 SQL을 다음의 [그림 2] 형식으로 바꾼다. 변환된 DFT_TABLE (n1, n2, n3, n4, n5, n6)의 값을 (v1, v2, v3, v4, v5, v6)라고 할 때 SQL 구문은 다음의 [그림 2]와 같다.

```
SELECT *
FROM DFT_TABLE p
WHERE NOT EXISTS (
            SELECT *
            FROM DFT_TABLE q
            WHERE
(q.n1 - v1)*(q.n1 - v1) + (q.n2 - v2)*(q.n2 - v2) +  (q.n3 - v3)*(q.n3 - v3) + (q.n4 - v4)*(q.n4 - v4) +  (q.n5 - v5)*(q.n5 - v5) + (q.n6 - v6)*(q.n6 - v6) <
(p.n1 - v1)*(p.n1 - v1) + (p.n2 - v2)*(p.n2 - v2) +  (p.n3 - v3)*(p.n3 - v3) + (p.n4 - v4)*(p.n4 - v4) +  (p.n5 - v5)*(p.n5 - v5) + (p.n6 - v6)*(p.n6 - v6) );
```

그림 1. 유사 검색을 위한 SQL correlated query
Fig 1. SQL correlated query for nearest neighbor

```
SELECT id
FROM DFT_TABLE
WHERE  n1 BETWEEN (v1 - r1) AND (v1 + r1)
    AND n2 BETWEEN (v2 - r2) AND (v2 + r2)
    AND n3 BETWEEN (v3 - r3) AND (v3 + r3)
    AND n4 BETWEEN (v4 - r4) AND (v4 + r4)
    AND n5 BETWEEN (v5 - r5) AND (v5 + r5)
    AND n6 BETWEEN (v6 - r6) AND (v6 + r6);
```

그림 2. 영역을 지정한 후 범위 검색 SQL
Fig 2. Range search SQL with region

[그림 2]의 SQL에서 r1, r2, r3, r4, r5, r6는 각 차원의 허용 범위 또는 오차 범위를 나타내므로 [그림 2]의 SQL은 유사범위 이내의 데이터를 찾아낸다. 검색 결과에서 가장 유사한 것을 찾는 것 또는 유사 정도에 따른 연산들은 후처리 과정에서 주기억장치에서 이루어지는 방법을 적용한다.

## 4. DFT를 이용한 유사검색 기능 평가

DFT로 변환된 데이터의 유사검색 기능 평가를 위하여 데이터는 무작위로 입력하였으며 다양한 경우의 시퀀스 패턴을 확인하기 위하여 [그림 3]과 같이 대표적인 패턴을 입력하였다. [그림 3]은 기본 데이터로 패턴별로 저장되어 있는 것을 보여주고 있으며, 데이터는 4개씩 패턴에 변화를 주어 입력하였고 시작하는 값은 각각 다르게 저장되어있다. ①번과 ⑦번은 급격하게 증가하는 값과 조금씩 증가하는 패턴을 정리하였고, ②번은 ①번과 ⑦번과 반대이며 대표적인 패턴들을 만들어 그 기능을 확인하였다.

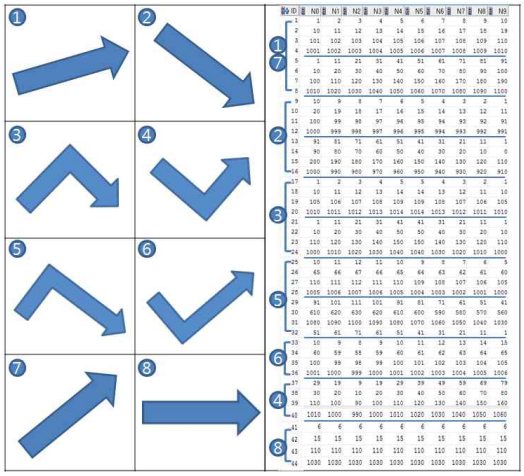

그림 3. 패턴별로 저장된 데이터
Fig. 3. Data assorted with various pattern

## 5. DFT를 이용한 유사검색 성능 평가

본 장에서는 DFT와 비트맵을 이용한 유사검색의 모의실험을 통하여 유사 검색의 성능을 평가하였으며 상황별로 그 결과를 설명하였다. [그림 2]의 영역을 지정한 유사 검색 SQL에서 유사범위를 나타내는 r의 값, 그리고 DFT로 변환한 후 사용할 속성의 개수 n에 따라 유사 검색 성능을 평가하였다.

### 5.1 구현 환경

[표 4]는 본 논문에서 사용한 유사 시퀀스 검색의 모의실험 구현환경이다.

표 4. 구현 서버와 클라이언트 환경
Table 4. Client and server configuration

| 서버환경 | |
|---|---|
| Red hat Linux 5.1.19.6 starting OS | 오라클 11g 서버 |

| 클라이언트환경 |
|---|
| CPU : Intel(R) Core(TM) i5 CPU 650 @ 3.20GHz 3.20 GHz |
| RAM : 4.00GB(3.05GB 사용가능) |

### 5.2 성능평가 기준측정

성능 평가의 기준은 DFT로 변환된 값의 개수와 값의 범위에 따른 비교방법과 결과의 수, 비용 등을 토대로 작성을 하였다. 데이터는 10,000개, 50,000개, 100,000개의 경우에 따른 결과로 실험을 하였으며10 경우에 따라 비교되는 실수, 허수의 개수 역시 변화되었으며 각각의 결과와 비용이 다른 경우를 보였으며 비트맵 인덱스를 사용하여 검색 속도를 높였다.

표 5. 비교될 값에 대한 범위
Table 5. Ranges of simulation data

| **비교될 값**의 앞 10%, 뒤 10%, **총 범위20%**, 데이터 : 100,000개(10만개) | | | |
|---|---|---|---|
| REAL_1 | -87.15 (앞 10%) | **-83 (비교될 값)** | -78.85 (뒤 10%) |
| REAL_2 | -355.95 | **-339** | -322.05 |
| REAL_3 | 263.15 | **277** | 290.85 |
| REAL_4 | 136.8 | **144** | 151.2 |
| REAL_5 | -1085.7 | **-1034** | -982.3 |
| REAL_6 | 136.8 | **144** | 151.2 |
| IMAG_1 | -808.5 | **-770** | -731.5 |
| IMAG_2 | 185.25 | **195** | 204.75 |
| IMAG_3 | 226.1 | **238** | 249.9 |
| IMAG_4 | -730.8 | **-696** | -661.2 |
| IMAG_5 | 0 | **0** | 0 |
| IMAG_6 | 661.2 | **696** | 730.8 |

| **비교될 값**의 앞 15%, 뒤 15%, **총 범위30%**, 데이터 : 100,000개(10만개) | | | |
|---|---|---|---|
| REAL_1 | -95.45 (앞 15%) | **-83 (비교될 값)** | -70.55 (뒤 15%) |
| REAL_2 | -389.85 | **-339** | -288.15 |
| REAL_3 | 235.45 | **277** | 318.55 |
| REAL_4 | 122.4 | **144** | 165.6 |
| REAL_5 | -1189.1 | **-1034** | -878.9 |
| REAL_6 | 122.4 | **144** | 165.6 |
| IMAG_1 | -885.5 | **-770** | -654.5 |
| IMAG_2 | 165.75 | **195** | 224.25 |
| IMAG_3 | 202.3 | **238** | 273.7 |
| IMAG_4 | -800.4 | **-696** | -591.6 |
| IMAG_5 | 0 | **0** | 0 |
| IMAG_6 | 591.6 | **696** | 800.4 |

| **비교될 값**의 앞 25%, 뒤 25%, **총 범위50%**, 데이터 : 100,000개(10만개) | | | |
|---|---|---|---|
| REAL_1 | -103.75 (앞 25%) | **-83 (비교될 값)** | -62.25 (뒤 25%) |
| REAL_2 | -423.75 | **-339** | -254.25 |
| REAL_3 | 207.75 | **277** | 346.25 |
| REAL_4 | 108 | **144** | 180 |
| REAL_5 | -1292.5 | **-1034** | -775.5 |
| REAL_6 | 108 | **144** | 180 |
| IMAG_1 | -962.5 | **-770** | -577.5 |
| IMAG_2 | 146.25 | **195** | 243.75 |
| IMAG_3 | 178.5 | **238** | 297.5 |
| IMAG_4 | -870 | **-696** | -522 |
| IMAG_5 | 0 | **0** | 0 |
| IMAG_6 | 522 | **696** | 870 |

[표 5]에서 비교될 값에 대한 범위는 기준 값의 10% (r=0.1), 15% (r=0.15), 25% (r=0.25) 범위로 정하였으며

표에 나타난 값은 각 기준 값 비율에 따른 실제 실수와 허수 값에 대한 범위이다.

[그림 4]를 보면 범위에 포함되는 결과 값의 개수가 작은 경우(사각형의 범위 내부)는 속성 개수를 각각 3개(실수, 허수)만 해도 충분한 결과 값을 얻을 수 있으나 범위에 속하는 결과 값이 많은 경우에는 많은 속성을 사용하는 것이 결과로 반환되는 결과 레코드 개수를 줄일 수 있다. [그림 5]를 보면 결과 값의 개수가 작은 경우(사각형 내부)는 속성 개수를 각각 3개(실수, 허수)만 해도 충분한 결과 값과 비용을 얻을 수 있었다. 이 경우 속성의 개수를 4, 5, 6으로 계속 늘려도 검색의 성능을 향상 시킬 수 없었다. 반면에 영역을 지정한 유사 검색의 결과에 범위에 속하는 값이 많은 경우에는 많은 속성을 사용하는 것이 더 정확한 검색에 의해 검색 비용이 줄어든 것을 볼 수 있다. 이 결과는 기존의 많은 연구 결과에서 단순히 DFT 변환 후 단순히 3개 또는 4개의 속성만을 사용하는 방법을 주장하고 있던 것과는 매우 다른 연구결과이다. 이는 기존의 연구 결과들이 유사 검색에서 유사 범위를 지정하는 방법을 사용하지 못하고 단순히 가장 유사한 것을 검색하는 방법을 적용하면서 검색 시 속성의 개수를 늘리면 비용도 늘어날 것으로 예측했기 때문이다.

| 데이터 수 | 10,000 | 50,000 | 100,000 | 100,000 | 100,000 | 100,000 |
|---|---|---|---|---|---|---|
| 속성3결과수 | 26 | 136 | 267 | 2 | 27 | 350 |
| 속성4결과수 | 17 | 88 | 154 | 1 | 9 | 181 |
| 속성5결과수 | 4 | 31 | 48 | 1 | 7 | 173 |
| 속성6결과수 | 2 | 13 | 19 | 1 | 7 | 173 |
| 범위 | ↑50%이상 | ↑50%이상 | ↑50%이상 | 범위20% | 범위30% | 범위50% |

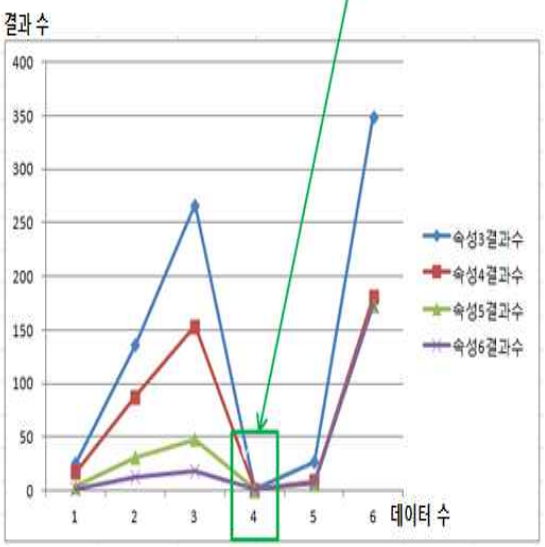


그림 4. 속성개수, 데이터개수, 범위에 따른 결과 그래프

Fig. 4. Graph of results with the changes of attribute, number of data and region

### 5.3 성능 평가 결과

본 논문의 실험결과에 의하면 유사 검색의 과정에서 유사도 (r의 값)를 사용한 검색 방법에서 검색 결과에 포함되는 결과의 개수가 적은 경우에는 실수, 허수의 속성개수를 기존 연구의 결과와 같이 각각 3개 정도만 해도 충분한 결과를 얻을 수 있었다. 하지만 속성을 3개 사용하여 유사 검색의 결과에 긍정 오류에 의해 반환되는 결과 값이 많다면 많은 비용이 필요하게 된다. 이는 비트맵을 사용함으로써 검색에 필요한 비용보다 반환되는 결과가 비용에 더 큰 비중을 차지하게 되어 발생하는 현상이다. 반환되는 결과가 많은 경우 속성의 개수를 추가하여 반환되는 결과 개수를 줄여 그에 따르는 비용 역시 줄여야 하겠으며 후처리 과정에 필요한 비용도 줄여 보다 정확한 결과를 얻을 수 있다. 결과 값을 보면 알 수 있듯이 속성의 개수가 3개만 해도 되는 경우가 있는가 하면 범위에 포함되는 결과 값이 많은 경우에는 비트맵 인덱스를 사용함으로 속성의 개수를 많이 하면 할수록 빠르고 정확한 결과를 얻을 수 있다. 필요에 따라 결과 값이 작다면 속성의 개수를 3개만 사용해도 기존 연구 결과와 같이 충분한 검색 결과를 얻을 수 있을 것이나 반면 유사 범위의 선택 폭이 너무 느슨해 결과의 개수가 많다면 속성의 개수 역시 4, 5, 6으로 늘려 반환되는 결과 값의 수도 줄이고 더불어 비용 역시 감소시킬 수 있다.

| 데이터 수 | 10,000 | 50,000 | 100,000 | 100,000 | 100,000 | 100,000 |
|---|---|---|---|---|---|---|
| 속성3비용 | 30 | 103 | 302 | 66 | 130 | 236 |
| 속성4비용 | 29 | 101 | 295 | 52 | 107 | 190 |
| 속성5비용 | 21 | 66 | 204 | 53 | 91 | 191 |
| 속성6비용 | 21 | 66 | 204 | 53 | 91 | 190 |
| 범위 | ↑50%이상 | ↑50%이상 | ↑50%이상 | 범위20% | 범위30% | 범위50% |

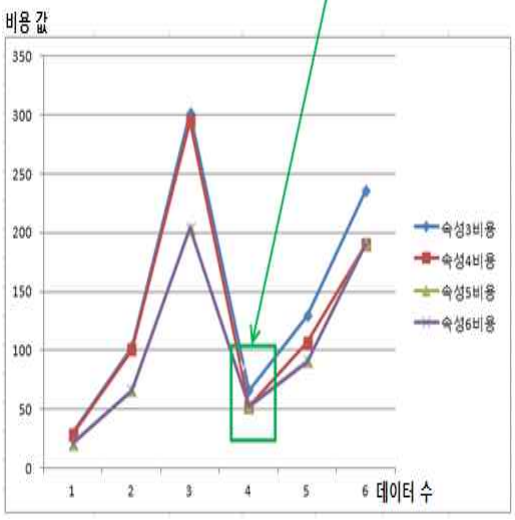


그림 5. 속성개수, 데이터개수, 범위에 따른 검색 비용 그래프

Fig. 5. Graph of Costs with the changes of attribute, number of data and region

## 6. 결론 및 향후 연구 방향

최근 대용량 데이터웨어하우스에 저장된 시퀀스에서 유사한 패턴을 가진 시퀀스 데이터를 효율적으로 검색하기 위한 연구가 많이 진행되어 왔다. 지금까지의 연구 방법들이 시퀀스 데이터를 주파수 영역의 데이터로 변환한 후 검색에 사용할 속성의 개수를 3개 또는 4개로 제한한 후 R-tree와 같은 공간 인덱스를 사용하는 방법을 적용해 왔다. 이 방법들은 공간 인덱스의 한계 문제로 속성의 개수를 3개 또는 4개로 제한해야 하는 단점이 있었다. 본 논문에서의 연구 방법은 영역 지정 방식의 질의 방식과 비트맵 인덱스를 사용하여 기존의 연구 결과들과 다른 실험결과를 보였다. 영역 지정 시 사용하는 유사도에 따라 질의 결과의 개수가 바뀜에 주파수 영역으로 변환된 데이터의 속성이 가변적으로 변동될 수 있음을 보였다.

본 논문에서의 연구 진행 방향은 다음과 같은 방법을 사용하였다. (1) 대용량 시퀀스 데이터의 확보, (2) 시퀀스에서의 불완전 데이터 보완, (3) DFT를 이용한 시퀀스 데이터의 변환 후 데이터웨어하우스에 저장, (4) 영역 지정 방식의 질의를 사용한 유사 검색의 성능 향상을 위한 비트맵 인덱스 생성, (5) 유사도의 지정과 비트맵 인덱스를 사용한 질의의 실행, (6) 상황에 따른 비교될 속성개수 변경 (7) 최소비용확인을 통한 유사 시퀀스 검색으로 연구를 진행하였다.

본 논문에서의 모의실험 결과에 의하면 대용량 시퀀스에서 유사 시퀀스를 검색하는 방법으로 시퀀스의 DFT 변환, 유사도 범위를 지정하는 방식의 질의, 그리고 비트맵 인덱스를 사용하는 실행 환경이 우수한 성능을 나타내는 것을 보였다. 이는 대부분의 상용 DBMS에서 사용하는 R-tree 또는 Quad 트리와 같은 공간 인덱스가 실제 환경에서 4차원 공간 이상을 지원하지 못하는 문제점으로 인하여 시퀀스를 주파수 영역으로 변환하여 사용하는 방식에서 2개 이상의 속성 (실수 2개, 허수 2개)을 사용하지 못하는 매우 실질적인 문제점을 해결하였다.

## 참 고 문 헌


[1] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, "Fast Subsequence Matching in Time-Series Databases," Proc. ACM SIGMOD, Minneapolis MN, May 25-27, pp. 419-429, 1994.

[2] Tak-chung Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence,* vol. 24, pp. 164-181, 2011.

[3] Maria Kontaki, Apostolos N. Papadopoulos, Yannis Mannolopoulos, "Adaptive similarity search in streaming time series with sliding windows," *Data and Knowledge Engineering*, vol. 63(2), pp. 478-502, 2007.

[4] V.Gaede, O.Gunther, "Multidimensional Access methods," *ACM Computing Surveys*, vol. 30(2), pp. 170-231. 1998.

[5] CS Perng, H. Wang, SR Zhang, DS Parker, "Landmarks: A new model for similarity-based pattern querying in time series databases," *Proc. of ICDE'00, San Diego*, pp. 33-42, CA, 2000.

[6] B. Yi, HV Jagadish, C. Faloutsos, "Efficient Retrieval of similar time sequences under time warping," *Proc. of ICDE'98, Orlando, FL*, pp. 201-208, 1998.

[7] R. Agrawal, K. Lin, H. Sawhney, and K.Shim, "Fast similarity search in the presence of noise, scaling, and translation in time-series databases," *Proc. of VLDB'95, Zurich, Switzerland*, pp. 490-501, 1995.

[8] Y. Moon, K. Whang, W. Loh, "Duality-based subsequence matching in time-series databases," *Proceedings of the 17th IEEE International Conference on Data engineering*, pp. 263-272, 2001.

[9] Y. Moon, K. Whang, W. Han, "General match : subsequence matching method in time-series databases based on generalized windows," *Proceedings of ACM SIGMOD*, pp. 382-393, 2002.

[10] E. Keogh, J. Lin, J. Fu, "HOT SAX: efficiently finding the most unusual time series subsequence," *Proceedings of the Fifth IEEE International Conference on Data Mining*, pp. 226-233, 2005.


## 저 자 소 개


**손동원(Dong-won Son)**

2010년 : 계명대학교 컴퓨터공학과 공학사.
2012년 : 계명대학교 대학원 컴퓨터공학 석사

관심분야 : 데이터베이스, 데이터마이닝, 질의최적화
Phone : 017-450-7889
E-mail : dwson@kmu.ac.kr, sdw7889@naver.com

**홍동권(Dong-Kweon Hong)**

1985년 : 경북대학교 전자과 공학사
1992년 : U. of Florida 컴퓨터공학 석사
1995년 : U. of Florida 컴퓨터공학 박사
1997년~현재 : 계명대학교 컴퓨터공학과 교수

관심분야 : XML, 데이터베이스, 질의 최적화
Phone : 053-580-5281
Fax : 053-580-5165
E-mail : dkhong@kmu.ac.kr