추가전용 데이터베이스에 대한 연속 마이닝

김룡[0] 이준욱 이양우 류근호
충북대학교 데이터베이스 연구실
(kimlyong[0], junux, dooji, khryu)@dblab.chungbuk.ac.kr

Continuous Mining Over Append-Only Databases

Long Jin[0] Jun Wook Lee Yang Woo Lee Keun Ho Ryu
Database Laboratory, Chungbuk National University

요 약

최근에 많은 새로운 타입의 어플리케이션에서 정보 시스템들에 대한 사용의 증가로 인해 연속 질의들은 여러 연구 프로젝트들에서 초점이 되고 있으며, 연구가 활발히 진행되고 있다. 특히 시계열에 대해서 미래의 값에 대한 예측 모델과 FFT(Fast Fourier Transform)을 이용하여 새로운 값이 입력될 때마다 신속하게 응답할 수 있는 이웃에 관한 연속 질의에 대해 이미 연구되었다. 그러나 이것은 이웃에 관한 질의이며 또한 방대한 데이터를 처리함에 있어서 매우 효율적이지 못하다. 이 논문에서는 시계열에 있어서 예측 모델을 이용하여 미래의 값을 예측한다. 다음 DFT(Discrete Fourier Transform)을 이용하여 변환한 후 R*-tree를 구성하고, 새로운 값이 입력될 때마다 신속하게 유사성 시계열들을 찾아서 응답하는 연속 범위 질의 과정과 시스템 구조에 대해 제안한다.

1. 서 론

최근에 많은 새로운 타입의 어플리케이션에서 정보 시스템들에 대한 사용의 증가로 인해 방대한 양의 스트림 데이터들이 생성되고 있다. 따라서 이들을 효율적으로 처리하기 위한 연속 질의에 관한 연구가 많이 진행되고 있다. 예로서, 주식 어플리케이션에서 시스템은 주식에 대해 질의하면, 값들이 미리 선택된 값들의 계열들과 유사한 추세를 보일 때마다 신속하게 응답하게 된다. 연속 질의의 예로, "주식A에 대해서 앞으로 일주일동안 이 주식과 유사한 패턴을 가진 주식을 찾아라."와 같다. [2]에서는 시계열에 대해 미래의 값에 대한 예측 모델과 FFT(Fast Fourier Transform)을 이용하여 새로운 값이 입력될 때마다 신속하게 응답할 수 있는 이웃(neighbor)에 관한 연속 질의(continuous query)를 제안하였다. 그러나 이것은 이웃에 관한 질의이며 또한 방대한 양의 데이터를 처리함에 있어서 질의와 스캔된 시계열들에 대한 거리를 전부 계산하였다. 그리고 이들에 대해 Quick Sort 알고리즘을 사용하여서 처리하기 때문에 매우 효율적이지 못하다. 따라서 이 논문에서는 연속 범위 질의를 효율적으로 처리하기 위한 방법과 이를 위한 시스템 구조를 제안한다. 즉, 시계열에 있어서 예측 모델을 이용하여 미래의 값을 예측하고, DFT(Discrete Fourier Transform)을 이용하여 변환한 후 R*-tree를 구성한다. 다음 질의와 스캔된 시계열들 사이의 거리를 전부 계산하는 것이 아니라, 질의에 대하여 R*-tree를 검색하면서 인덱스에 대한 상한과 하한을 이용하여 해당하는 시계열들만 검색하여서 그것에 대한 거리를 계산한다. 또한 질의 결과 집합들을 생성한 후 새로운 값이 입력될 때마다 검증을 거쳐서 신속하게 유사성 시계열들을 찾아서 응답하게 된다.

이 논문의 전체적인 구성은 다음과 같다. 2절에서는 관련연구를 기술하였으며, 3절에서는 예측처리 과정에 대해서 기술한다. 제4절에서는 연속적인 유사성 계열을 검색하는 과정에 대해 제시한다. 그리고 5절에서는 연속 범위 질의 과정에 대한 시스템 구조에 대해 다루며, 마지막으로 6절에서는 이 논문의 결론을 내린다.

2. 관련연구

시계열 데이터베이스에서 여러 가지 효과적인 유사성 검색 기법들이 많이 연구되었다. [3]에서는 시계열 데이터를 시간 도메인에서 빈발 도메인으로 매핑하는 DFT(Discrete Fourier Transform)을 이용한 전체 매칭에 대해 제안하였다. 일부분의 빈발 계수를 제외한 기타 모든 계수들을 제거하고 취한 계수들을 R*-tree와 같은 다차원 인덱스 구조를 이용하여 인덱스 한다. 그러나 [3]에서의 접근법은 데이터 시퀀스와 질의 시퀀스의 길이가 같아야 된다는 제한점을 갖고 있다. 이러한 단점을 해결하기 위한 방법으로서 서브 시퀀스 매칭에 대해 [4]에서 제안하였으며, 데이터 시퀀스에 대하여 슬라이딩 윈도우를 사용한 서브 시퀀스의 매칭을 허용한다. 즉, 매개 윈도우를 DFT를 사용하여 빈발 도메인으로 매핑한 후, 단지 일부분의 계수만 취한다. 하나의 시퀀스는 요소 공간에서 하나의 궤적(trail)으로 매핑한다. 이러한 궤적들은 MBR들에 의하여 서브 궤적들로 분할하여 인덱싱을 위한 R*-tree에 저장한다. 또한 [5]에서 앞의 DFT를 더욱 효과적으로 신속하게 처리하기 위한 방법으로 FFT를 제시되었다.

연속적인 질의에 있어서 [1]에서는 연속 개념과 이것을 기반으로 한 연속 질의 과정에 대해서 정의하였다. 즉, 이 논문에서는 연속 질의에 있어서 질의를 단순 질의(monotone query)로 변환하고, 이것을 다시 점진적인 질의(increment query)로 변환하는 연속 질의 과정에 대해 제시하였다. [2]에서는 시계열에 대해서 미래의 값에 대한 예측 모델과 FFT을 이용하여, 새로운 값이 입력될 때마다 신속하게 응답할 수 있는 이웃에 관한 연속 질의에 대해 제안하였다.

3. 예측 계열

이 절에서는 기본적인 수식과 연속 마이닝 과정에 대해서 설명한다. 먼저 이 논문에서 사용되는 심벌과 예측 계열에 대해서 표1와 같다.

표1 사용되는 심벌들과 의미

심벌	의미
x, y, ...	Time series
x[i,j]	x의 i와 j 사이의 서브 계열
LS	스트림 계열
PS	예측 계열
F_i	i번째 패턴 혹은 요소 계열로서 길이는 $l_i + 1$

모든 시계열들이 균일한 시간 간격에 따라 샘플링 되었다고 가정하자. 또한 모든 시계열들은 샘플링 시간에 따른 시퀀스들의 시점에 대해서 실수들의 시퀀스로 표현되었다. 이제부터 첫번째 샘플은 언제나 0으로부터 시작한다고 가정하자. 그러므로 시계열 x는 <x[0], x[1], ..., x[l],..,>와 같은 폼을 가진다. x는 $l \geq 0$으로 끝나는 유한한 시계열이라고 하면, 이 계열은 $l+1$의 길이를 가진다. 만약 이와 같이 l이 존재하지 않으면, 이 시계열은 무한하다. 만약 x가 시계열이라고 하면, x[i,j]을 사용하여 유한한 시계열 <x[i], x[i+1], ..., x[j]>를 나타낸다. 여기서 0≤i≤j인 정수이고, j는 x의 길이보다 작다.

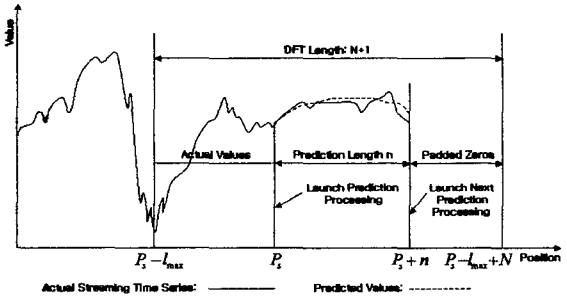

그림 1 예측 처리과정

현재 시점 p_s라고 하면, 연속 질의는 시점 p_s-1를 포함하고, 시점 p_s에 대한 값은 아직 입력되지 않은 상태에서 응답하여야 한다. 이 경우, n 단계 예측을 통하여 하며, 이것이 그림1에 묘사되었다. p_s 시점 다음에 대해서 예측 모델은 시점 p_s, p_s+1, ..., p_s + n −1에 있어서 n 단계의 예측 값을 얻는다(점선으로 된 부분). PS는 p_s-1보다 작은 규격화된 시계열 LS, n 예측 값들, 무한한 0값들로 구성되었다. PS를 예측 시계열이라고 한다. 상세히 나타내면,

PS = <LS[0], ..., LS[p_s-1], P_0, ..., P_{n-1}, 0, ..., 0, ...>

여기서 P_i, i = 0, ..., n−1 은 예측 값이다.

이와 같은 PS가 다음절에서 소개할 두 시계열 사이의 거리를 계산할 때 실제적으로 사용한다.

4. 연속 서브 시퀀스 유사성 검색(연속 마이닝)

[4]에 의하여 길이 N+1를 가진 두 계열 x와 X가 주어졌을 때, X_{DFT} 은 X가 x에 대한 (N+1)-point DFT임을 나타낸다. 정확하게, 만약 이 경우, x는 X에 대한 (N+1)-point DFT의 역함수이다. 길이 N+1를 가진 시계열 y와 Y도 위와

$$D(x,y) = \sqrt{\sum_{i=0}^{N} (X[i]-Y[i])^2} \quad \cdots\cdots (1)$$

마찬가지라고 하면, 이 두 시계열 사이의 거리는 다음과 같다:

거리에 대한 3자 관계에 대해서 [2]에 나와 있다. 간단히 설명하면 다음과 같다. 시점 p를 고려하면, 이것은 p_s 에서 p_s + n −1 사이이다. 3절에서 논의하였듯이 예측을 통하여 값들을 예측한 후 수식 (1)을 통하여 매개 F_i에 대한 예측 거리 $D(PS[p-l_i,p],F_i)$ 는 이미 알고 있다. 언제나 시점 p에서 길이 l_i+1를 가진 예측 계열들과 실제적인 시계열 LS 사이의 예측 에러를 점진적으로 쉽게 계산할 수 있다. 시점 p에 대한 LS의 데이터 값이 입력되면 $D(PS[p-l_i,p],LS[p-l_i,p])$ 값을 알게 된다. 위의 예측 거리와 예측 에러들에 의하여, 시점 p에서 실제적인 스트림 계열들과 매개 패턴 계열들에 대한 거리의 상한과 하한을 유도할 수 있다. 실제로 유클리디안 거리의 변화량을 사용하여 3자간의 관계를 얻을 수 있다:

$$|D(x,F_i)-D(x,y)| \leq D(y,F_i) \leq |D(x,F_i)+D(x,y)| \quad \cdots\cdots (2)$$

여기서 $x = PS[p-l_i,p]$와 $y = LS[p-l_i,p]$이다. 3자간의 관계를 그림 2으로 나타낼 수 있다.

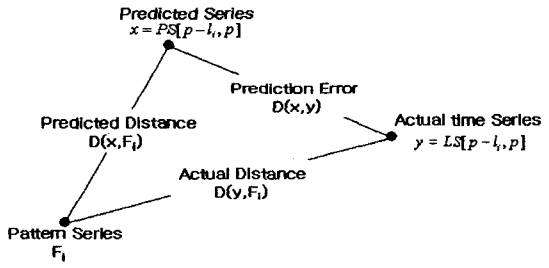

그림 2 거리에 대한 3자 관계

수식 (2)는 시점 p에 대한 매개 패턴 F_i에 대해 적용된다. 알고리즘을 간단히 하기 위하여 최대 예측 에러를 이용하여 위의 경계를 계산하며, 전체 l_i에 있어서,

$\max D(PS_p,LS_p) = \max\{D(PS[p-l_i,p],LS[p-l_i,p])\}$ 로 나타낸다. 주어진 F_i에 있어서, $D(PS[p-l_i,p],F_i)+\max D(PS_p,LS_p)$은 상한을 나타내고, $D(PS[p-l_i,p],F_i)-\max D(PS_p,LS_p)$은 하한을 나타낸다.

이것을 이용하여 다음 4.2절에서 연속적인 유사성 검색을 하게 된다.

4.2 연속 유사성 검색

수식 (2)에서 3자간의 관계에 대해서 매개 시점 p에서 세 개의 거리로서 그림 2에 잘 나와 있다. 이러한 3자간의 관계로부터 상한과 하한을 유도한다. 이것을 이용하여 R*-tree로부터 유사성 계열들을 찾아낸다. 그 구체적인 과정은 다음과 같다. (1) PS와 LS 사이의 예측 에러거리로부터 최대 예측 에러 거리 maxD를 구한다. (2) 다음 R*-tree에 대해서 질의를 검색할 때 매개 인덱스에 있어서 maxD를 이용하여 상한 거리와 하한 거리를 구할 수 있다. (3) 이때 ① 만약 상한 거리가 임계값 ε보다 작거나 같으면 이것은 유사성 계열로써 유사성 계열집합에 속하며, ② 만약 상한 거리가 임계값 ε보다 크고 하한 거리가 임계값 ε보다 작으면 이것은 후보집합에 속한다. ③만약 하한 거리가 임계값 ε보다 크면 이것은 추출집합에 속하게 되며 삭제하게 된다. 여기서 주의할 점은 유사성 계열 집합에 속한 계열도 오류기각(false alarm)을 갖고 있으므로 후처리 과정을

통하여 반드시 걸러내야 한다. 이 과정에 대해서 그림3와 같이 알고리즘으로 나와 있다.

INPUT OUTPUT	PS, T(R*-tree root point) Candidate patterns Set C and Similar patterns Sets R'
METHOD:	Step 1 Let $maxD(PS_p, LS_p) = max\{D(PS[p-l_i, p], LS[p-l_i, p])\}$ for all l_i. Step 2 For T, Finds all Similar patterns and candidates patterns. temporal R*-tree index argument *tem*, if($(D(PS, tem) + maxD) \le \varepsilon$) It adds to R' else if((($D(PS, tem) + maxD) \ge \varepsilon$) && ($D(PS, tem) - maxD) \le \varepsilon$)) It adds to C Step 3 Return Candidate patterns Set C and Similar patterns Sets R'

그림 3 P시점에서 유사성 패턴과 후보집합을 찾는 알고리즘

여기서 이 사용자 질의는 RuleSet에 저장된다. 위에서 구한 후보집합에 대해서 매번 새로운 값이 실제적으로 입력되면 RuleSet에 있는 이 질의가 후처리 과정에서 불려져서 후보집합을 검증하게 된다. 만약 후보집합에서 실제적으로 입력된 값에 의하여 실제적인 거리가 임계값 ε보다 작거나 같으면 이것은 유사성 계열의 집합에 속하게 된다. 이렇게 함으로써 새로운 값이 입력되면 연속적으로 신속하게 질의를 처리하여서 응답하게 된다. 이와 같이 전체적인 실제 값들이 전부 입력될 때까지 이 질의는 연속적으로 계속 반복하여서 진행하게 된다.

5. 시스템 구조

이제까지 위 부분에서 논의한 전체적인 과정에 대한 시스템의 구조가 다음 그림4 에 나와 있다.

인덱스 구축단계 (Index Creation Phase) 는 다음과 같은

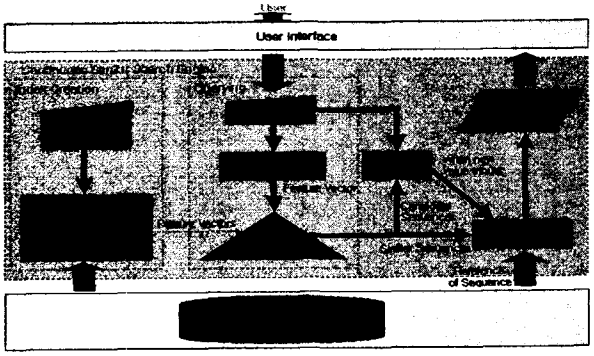

그림 4 시스템 구조

과정을 포함한다.

1. 원시 시퀀스 데이터를 취한 후 이것을 슬라이딩 윈도우를 사용하여 고정된 길이의 서브시퀀스로 자른다. 슬라이딩 윈도우의 크기를 ω 라고 하자. 그러면 원시 시퀀스의 전체 서브 시퀀스의 개수는 n-ω -1이다. 서브 시퀀스 S_i는 고정된 크기 ω 이다 ($0 \le i \le n-\omega-1$).
2. 모든 데이터가 한정된 영역 내에 있게 하기 위하여 모든 서브 시퀀스를 일반화(normalize)를 한다.
3. DFT를 이용하여 서브 시퀀스들을 빈발(frequency) 도메인으로 변환한다.
4. 단지 앞부분의 몇 개의 DWT 계수들을 사용하여 원시 서브시퀀스를 나타낸다.
5. 다차원 인덱스 구조로서 R*-tree를 사용하여 선택된 부분적인 계수들을 인덱스화 하고, 해당된 시계열들만 신속하게 검색할 수 있게 한다.

이를 기반으로 질의 단계(Querying Phase)는 다음과 같은 과정을 포함한다.

1. 인덱스 과정 2와 3과 같이 질의 Q에 대하여 일반화와 DFT 변환을 한다.
2. 인덱스 과정 4와 같이 질의 Q에 대하여 일부분의 DFT 계수를 취하여 Q' 를 얻는다.
3. 취한 일부분의 계수들을 이용하여 인덱스 구조에서 검색한다. 즉 인덱스 구조에서 Q' 로부터 거리가 ε내에 있는 모든 시퀀스들을 찾는다. 결과 R' 는 Q' 로부터 거리가 ε내에 있는 모든 시퀀스들의 집합과 후보집합 C를 얻는다.
4. R' 에 있는 매개 시퀀스에 대하여, 만약 Q로부터 거리가 ε내에 있는지를 후처리 과정에서 진행하며, 만족하면 이것들을 최종 결과 R에 넣은 후 이것을 사용자에게 반환한다
5. 질의 Q는 RuleSet에 저장된다.
6. 새로운 값이 입력되면 RuleSet에 있는 Q가 불리어지면서 후보집합 C에 대해서 후처리 과정에서 진행하며, 만약 거리가 ε내에 있는 시퀀스들이 있으면 이것들을 최종 결과 R에 추가한 후 R를 사용자에게 반환한다.

6. 결론

최근 연속 질의 연구에 대한 관심 증대되고 있다. 이 논문은 시계열에 대한 유사성 시계열을 찾는 연속 범위 질의에 대해서 다루었다. 전체적으로 이 논문에서는 시계열에 대해서 신속하게 유사성 시계열들을 찾아서 응답하기 위하여 미래의 값에 대해서 예측 모델을 이용하여 예측하였다. 다음 DFT를 이용한 거리에 대한 정의와 예측 계열에 대해서 정의하였다. 또한 효율적으로 처리하기 위하여 R*-tree를 이용하여 해당된 시계열을 신속하게 취하도록 하였다. 그리고 이들을 이용하여 새로운 값이 입력될 때마다 신속하게 유사성 시계열들을 찾아서 응답할 수 있는 연속 범위 질의 과정에 대해서 알아보았으며, 전체적인 시스템 구조에 대해서 재시하였다. 향후 연구로는 유사성 시계열들을 찾는 연속적인 질의에 대한 실제적인 시스템 구현과 성능에 대한 평가가 필요하다.

7. 참고문헌

[1] Douglas B. Terry, David Goldberg, David Nichols, Brian M. Oki: Continuous Queries over Append-Only Databases. SIGMOD Conference 1992, 321-330.

[2] Like Gao, Xiaoyang Sean Wang: Continually Evaluating Similarity-Based Pattern Queries on a Streaming Time Series. SIGMOD Conference 2002.

[3] R. Agrawal, C. Faloutsos, and A. Swami: Efficient Similarity Search in Sequence Databases. In Proceedings of the 4th Int'l Conference on Foundations of Data Organization and Algorithms, pp. 69-84, Chicago, Oct. 1993.

[4] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos: Fast Subsequence Matching in Time-Series Database. In Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, pp. 419-429, Minneapolis, May 1994.

[5] M. Frigo and S. G. Johnson. FFTW: C subroutine library for computing the Discrete Fourier Transform(DFT). On-line. http://www.fftw.org/, 2001.