시계열 서브시퀀스 매칭에서 GeneralMatch와 DualGmatch의 비교 분석

이상훈°, 문양세
강원대학교 컴퓨터과학과
{sanghun, ysmoon}@kangwon.ac.kr

A Comparative Analysis of GeneralMatch and DualGMatch in Time-Series Subsequence Matching

Sanghun Lee and Yang-Sae Moon
Dept. of Computer Science, Kangwon National University

요 약

최근 시계열 데이터베이스 기반의 다양한 응용 분야에서 서브시퀀스 매칭(subsequence matching) 연구가 활발히 진행되고 있다. FRM과 DualMatch은 효과적인 서브시퀀스 매칭을 위해 처음 제안된 해결책이다. 이후 이들을 일반화한 GeneralMatch가 제안되었으며, 최근에는 GeneralMatch의 이원적 접근법인 DualGMatch가 제안되었다. 본 논문에서는 GeneralMatch와 DualGMath를 비교 분석 하고자 한다. 이를 위해, 먼저 윈도우 구성 관점에서 GeneralMatch와 DualGMatch를 평가한다. 다음으로, 두 해결책을 최대 윈도우 크기 효과와 인덱스 저장 효율 관점에서 이론적으로 비교 분석한다. 마지막으로, 실제 시계열 데이터를 활용하여 GeneralMatch와 DualGMatch의 인덱스 페이지 접근 횟수를 비교한다. 분석 결과, GeneralMatch가 윈도우 크기 효과와 인덱스 저장 효율 측면에서 DualGMatch보다 우수한 것으로 나타났다.

1. 서론

시계열 데이터의 대표적인 예로는 주식, 센서, 온도, 의료 데이터 등이 있다[1, 2, 3, 4]. 시계열 데이터베이스에 저장된 시계열 데이터를 **데이터 시퀀스**(*data sequence*)라 부르며, 사용자에 의해 주어진 **질의 시퀀스**(*query sequence*)와 유사한 데이타 시퀀스를 검색하는 방법을 **유사 시퀀스 매칭**(*similar sequence matching*)이라고 한다[1, 5, 6]. 본 논문에서는 유클리디안 거리를 유사 모델(similarity measure)로 사용하는데, 길이 n 인 두 시퀀스 X(={$X[1]$, $X[2]$, …, $X[n]$})와 Y(={$Y[1]$, $Y[2]$, …, $Y[n]$})의 유클리디안 거리 $D(X, Y)$는 $\sqrt{\sum_{i=1}^{n}(X[i]-Y[i])^2}$으로 정의된다. **서브시퀀스 매칭**(*subsequence matching*)[1, 5, 7]은 질의 시퀀스와 유사한 서브시퀀스를 가지는 데이터 시퀀스와 해당 서브시퀀스의 위치를 찾는 문제이다.

일반적으로, 서브시퀀스 매칭은 인덱싱(indexing) 단계와 매칭의 두 단계로 구성된다[1, 5, 7]. 먼저, 인덱싱 단계에서는 데이터 시퀀스를 사이즈 ω의 윈도우(windows)로 나눈다. 그리고 나누어진 각 윈도우를 f-차원의 점으로 저차원 변환하여 인덱스에 저장한다. 다음으로, 매칭 단계에서는 질의 시퀀스를 사이즈 ω의 윈도우로 나눈다. 그리고 각 윈도우를 f-차원의 점으로 저차원 변환하고 인덱스를 검색하여 질의 시퀀스와 유사할 가능성이 높은 데이터를 후보로 삼는다. 마지막으로, 실제 데이터베이스를 액세스하여 이들 후보 중에서 실제로 질의 시퀀스와 유사한 서브시퀀스들만을 찾는다. 표 1은 윈도우를 사용하는 대표적인 네 가지 서브시퀀스 매칭 해결책을 나타낸다.

본 논문에서는 GeneralMatch와 DualGMatch를 경험에 기인하여 비교하고 분석한다. 참고문헌 [7]에서 언급한 바와 같이, 서브시퀀스 매칭 해결책으로 처음 제안된 FRM과 DualMatch는 점 여과 효과(point-filtering effect)의 결여와 작은 윈도우 크기 효과(window size effect)라는 단점을 각각 가지고 있다. 이후 이들의 단점을 보안하고자 GeneralMatch 해결책이 제안되었다. GeneralMatch는 데이터 시퀀스를 J-슬라이딩 윈도우(J-sliding window)로 나누고, 질의 시퀀스를 J -디스조인트 윈도우(J-disjoint window)로 나누는 방법을 사용함으로써, 점 여과 효과와 윈도우 크기 효과의 장점을 동시에 활용 가능한 해결책이다. 또한, 최근 제안된 해결책인 DualGMatch는 GeneralMatch 보다 우수한 성능을 보이기 위해 J-슬라이딩 윈도우와 J-디스조인트 윈도우를 선택적으로 적용한다. DualGMatch와 GeneralMatch는 FRM과 DualMatch의 일반화된 접근법으로, 이 둘의 비교 분석이

이 논문은 2014 년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2014R1A2A2A01002548)

표 1. 윈도우 구성 방법에 따른 서브시퀀스 매칭 해결책.

Solutions	Data sequence	Query sequence	Remark
FRM[1]	sliding window	disjoint window	First solution
DualMatch[5]	disjoint window	sliding window	Dual approach of FRM
GeneralMatch[7]	*J*-sliding window	*J*-disjoint window	Generalization of FRM and DualMatch
DualGMatch[8]	*J*-disjoint window	*J*-sliding window	Dual approach of GeneralMatch

필요하다. 따라서, 본 논문에서는 GeneralMatch와 DualGMatch를 소개하고 이들 매칭 방법을 경험에 기인하여 비교 분석하고 평가한다. 분석 결과, GeneralMatch는 DualGMatch보다 우수한 성능을 보이는 것으로 나타났다. 이는 GeneralMatch가 DualGMatch보다 큰 윈도우 크기 효과를 발휘함과 동시에 점 여과 효과를 충분히 활용하기 때문이다.

2. GeneralMatch와 DualGMatch

본 논문에서는 GeneralMatch와 DualGMatch의 설명을 위해 먼저 *J*-슬라이딩 윈도우와 *J*-디스조인트 윈도우에서의 윈도우를 정의한다.

정의 1: 길이 n인 시퀀스 $X = \{X[1], \ldots, X[n]\}$와 윈도우 사이즈 ω가 주어졌을 때, i-번째 *J*-슬라이딩 윈도우 x_i^J와 (J, k)-번째 *J*-디스조인트 윈도우 $x_{(j,k)}^J$를 서브시퀀스 $X[m : m + \omega - 1]$와 $X[n : n + \omega - 1]$라 정의한다. 이 때, $m = (i - 1) * J + 1$, $n = J + (k - 1) * \omega$이다. □

정의 1에 대한 보다 자세한 설명은 참고문헌 [7, 8]을 참조한다

2.1 GeneralMatch

GeneralMatch[5]는 데이터 시퀀스를 *J*-슬라이딩 윈도우로 나누고 질의 시퀀스를 *J*-디스조인트 윈도우로 나누는 방법이다. *J*-슬라이딩 윈도우로 나눈 데이터 시퀀스로는 인덱스를 구성하고, *J*-디스조인트 윈도우로 나눈 질의 시퀀스로 인덱스를 검색한다. 정리 1은 GeneralMatch의 정확성을 나타낸다.

정리 1: 데이터 시퀀스 S와 질의 시퀀스 Q를 각각 *J*-슬라이딩 윈도우와 *J*-디스조인트 윈도우로 나누었을 때, 두 시퀀스 $S[i : j]$와 Q가 ε-매치($D(S[i : j], Q) \le \varepsilon$)하면, 적어도 하나의 ($S[i : j]$의 *J*-슬라이딩 윈도우, Q의 *J*-디스조인트 윈도우)쌍이 $\varepsilon/\sqrt{\rho}$-매치한다. 이 때, ρ는 $S[i : j]$와 Q에 포함된 윈도우 개수이다. □

정리 1은 필요 조건을 만족하는 서브시퀀스 $S[i : j]$로 구성된 후보 집합이 착오기각(false dismissal)[1, 3]이 발생하지 않는다는 것을 보장한다. 정리 1에 대한 증명은 참고문헌 [5]를 참조한다.

표 1에서 보듯이, FRM과 DualMatch는 *J*-슬라이딩 윈도우와 *J*-디스조인트 윈도우를 사용하는 대표적인 해결책이다. FRM의 슬라이딩 윈도우는 1-슬라이딩 윈도우에, DualMatch의 디스조인트 윈도우는 ω-디스조인트 윈도우에 해당한다. FRM이나 DualMatch와는 달리 GeneralMatch는 일반화된 윈도우를 사용함으로써, 특수한 경우인 FRM과 DualMatch 모두에 적용이 가능하다. 즉, GeneralMatch의 윈도우 구성 시 J를 1로 사용하면 FRM과 동일하고, J를 ω로 사용하면 DualMatch와 동일한 효과를 나타낸다.

2.2 DualGMatch

DualGMatch는 GeneralMatch의 이원적 접근법으로, 데이터 시퀀스를 *J*-디스조인트 윈도우로 나누고 질의 시퀀스를 *J*-슬라이딩 윈도우로 나누는 방법을 사용한다. 정리 2는 이러한 DualGMatch의 정확성을 나타낸다.

정리 2: 데이터 시퀀스 S와 질의 시퀀스 Q를 각각 *J*-디스조인트 윈도우와 *J*-슬라이딩 윈도우로 나누었을 때, 두 시퀀스 $S[i : j]$와 Q가 ε-매치($D(S[i : j], Q) \le \varepsilon$)하면, 적어도 하나의 ($S[i : j]$의 *J*-디스조인트 윈도우, Q의 *J*-슬라이딩 윈도우)쌍이 $\varepsilon/\sqrt{\rho}$매치한다. 이 때, ρ는 $S[i : j]$와 Q에 포함된 윈도우 개수이다. □

DualGMatch는 GeneralMatch의 이원적 접근법이다. 따라서, 정리 2는 정리 1과 윈도우 구성 방법의 이원적 접근으로 후보 데이터 집합에 대해 착오기각이 발생하지 않는다는 것을 간단하게 증명 가능하다.

GeneralMatch에서와 같이 DualGMatch 역시 특수한 윈도우를 사용한 경우인 FRM과 DualMatch에 적용이 가능하다. 즉, 데이터 시퀀스를 ω-슬라이딩 윈도우로 사용하면 FRM, 디스조인트 윈도우를 1-디스조인트 윈도우로 사용하면 DualMatch와 동일한 효과를 나타낸다.

3. 분석 평가

서브시퀀스 매칭의 성능은 (1) 윈도우 크기 효과와 (2) 점 여과 효과[5]에 따라 크게 달라진다. 먼저, 윈도우 크기 효과를 크게 하기 위해서는 가능한 한 큰 윈도우를 사용해야 한다. 따라서, 본 논문에서는 윈도우 크기에 따른 서브시퀀스 매칭 성능을 분석하고자 한다. 둘째, 점 여과 효과를 최대로 활용하기 위해서는 점(데이터)을 인덱스에 직접 저장해야 하는데, 이 때 모든 점들은 최소 경계 사각형(minimum bounding rectangles: MBRs)[3]의 구성 없이 인덱스에 저장된다고 가정한다. 따라서, 인덱스에 저장된 점은 서브시퀀스 매칭의 비교에서 중요한 척도이며, 본 논문에서는 저장된 점의 수에 따른 서브시퀀스 매칭 성능을 분석한다. 표 2는 네 개의 서브시퀀스 매칭 해결책에서 인덱스에 저장된 점의 수와 최대 윈도우 크기를 결정하는 방법을 나타낸다[1, 4, 7, 8].

표 2. 네 가지 해결책에서 최대 윈도우 크기와 점의 수 결정 방법.

Solutions	Maximum window size	Number of points	Remark
FRM[1]	$Min(Q)$	$Len(S) - \omega + 1$	$J = 1$ in GeneralMatch
DualMatch[5]	$\left\lfloor \frac{Min(Q)+1}{2} \right\rfloor$	$\left\lfloor \frac{Len(S)}{\omega} \right\rfloor$	$J = 1$ in DualGMatch
GeneralMatch[7]	$\left\lfloor \frac{Min(Q)-J+1}{2} \right\rfloor * J$	$\left\lfloor \frac{Len(S)-\omega}{J} \right\rfloor + 1$	$J = 1, \ldots, \omega$
DualGMatch[8]	$\left\lfloor \frac{Min(Q)+J}{2J} \right\rfloor * J$	$\sum_{i=1}^{J} \left\lfloor \frac{Len(S)-i+1}{\omega} \right\rfloor$	$J = 1, \ldots, \omega$

데이터 시퀀스와 질의 시퀀스에 대해 인덱스에 저장된 점의 수와 최대 윈도우 크기는 표 2를 사용하여 계산된다. 본 논문에서는 보다 명확한 분석을 위해 질의 시퀀스의 길이는 512로($Len(Q) = 512$), 데이터 시퀀스의 길이는 1,000,000($Len(S) = 1{,}000{,}000$)으로 고정하여 사용하였다. 그림 1은 인덱스에 저장된 점의 수에 따른 GeneralMatch와 DualGMatch의 최대 윈도우 사이즈를 나타낸다. 그림에서 보듯이, 점의 개수가 3,906인 경우 GeneralMatch와 DualMatch가 동일한 최대 윈도우 크기를 가지며, 점의 개수가 999,489인 경우 DualGMatch와 FRM이 각각 동일한 최대 윈도우 크기를 갖는 것을 알 수 있다. 이는 앞서 설명한 대로 DualGMatch와 GeneralMatch가 일반화된 윈도우를 사용하기 때문이다. 즉, 질의 시퀀스의 길이가 512인 경우, 표 2의 식을 통해 계산된 DualMatch와 FRM의 최대 허용 윈도우 크기는 각각 256과 512로, 이는 GeneralMatch와 DualGMatch의 특수한 경우 최대 윈도우 크기와 동일하다. 이 특수한 두 경우를 제외하고는 동일한 점의 수에 대해 GeneralMatch의 최대 윈도우 크기는 DualGMatch보다 큰 것을 알 수 있다. 결론적으로, 인덱스에 동일한 점이 저장된 경우 GeneralMatch는 DualGMath보다 큰 최대 윈도우 크기를 가지며, 이에 따라 GeneralMatch는 DualGMatch보다 좋은 성능을 나타낸다.

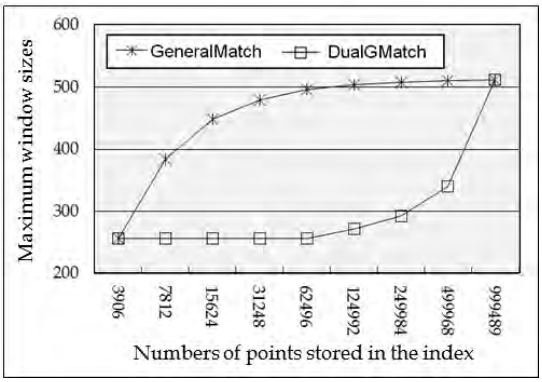

그림 1. 인덱스에 저장된 점의 수에 따른 최대 윈도우 크기 변화.

그림 2는 동일한 윈도우 크기에 대해 인덱스에 저장된 GeneralMatch와 DualGMatch의 점의 수를 나타낸다. 그림을 보면, 윈도우 사이즈가 256, 512인 경우 GeneralMatch와 DualGMatch는 동일한 점의 수를 갖는다. 이 두 경우를 제외하고는 동일한 윈도우 크기에 대해 DualGMatch는 GeneralMatch보다 항상 많은 수의 점을 인덱스에 저장한다. 즉, DualGMatch는 GeneralMatch보다 인덱스 검색에서 많은 오버헤드를 발생시고, 이는 성능 저하로 나타난다.

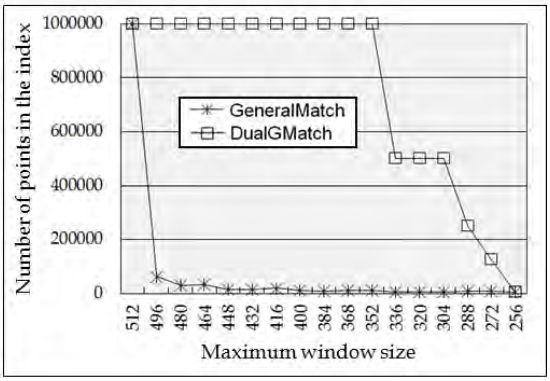

그림 2. 최대 윈도우 크기에 대해 인데스에 저장 가능한 점의 수 변화.

그림 1과 2에서 GeneralMatch가 DualGMatch보다 우수한 결과를 보이는 이유는 다음과 같다. 제2절에서 설명했듯이, GeneralMatch는 데이터 시퀀스를 J-슬라이딩 윈도우로 나누고 DualGMatch는 J-디스조인트로 윈도우로 데이터 시퀀스를 나눈다. 윈도우 크기가 동일한 경우, 시퀀스를 J-디스조인트 윈도우로 나누지 않고 J-슬라이딩 윈도우로 나누면, 보다 적은 수의 점이 생성된다(표 2 참조). 따라서 GeneralMatch는 DualGMatch보다 적은 수의 점을 생성하게 된다. 또한, GeneralMatch는 DualGMatch보다 인덱스에 동일한 수의 점을 저장하기 위해 보다 큰 윈도우 크기를 갖는다. 이와 반대로, 질의 시퀀스의 경우 GeneralMatch는 J-조인트 윈도우를, DualGMatch는 J-슬라이딩 윈도우를 사용하여 나누기 때문에 GeneralMatch가 보다 많은 수의 점을 생성한다. 하지만 질의 시퀀스로에서 생성된 점은 메인 메모리에서 처리할 수 있는 수준으로 매우 적은 수이며, 이는 전체 성능에 거의 영향을 미치지 않는다. 결과적으로, GeneralMatch는 DualGMatch보다 인덱스에 적은 수의 점만을 저장하고 보다 큰 최대 윈도우를 사용하여 우수한 성능을 나타낸다.

4. 결론 및 향후 연구

본 논문에서는 네 가지 서브시퀀스 매칭 솔루션인 FRM, DualMatch, GeneralMatch, DualGMatch를 소개하였다. 또한, 이 중 최근에 발표된 GeneralMatch와 DualGMatch를 최대 허용 윈도우 크기와 인덱스에

저장되는 점의 개수를 중심으로 비교 분석하였다. 분석 결과, 동일한 윈도우 크기에 대해 GeneralMatch가 DualGMatch 보다 인덱스에 저장되는 점의 개수는 적으며, 동일한 점의 개수를 저장하는 윈도우 크기는 더 큰 것으로 나타났다. 향후 연구로는 유사 거리 모델인 DTW(dynamic time warping)와 유클리디안 거리를 비교 분석할 예정이다. 또한, GeneralMatch와 DualGMatch의 최적 *J*값을 연구할 예정이다.

참고문헌

[1] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "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.

[2] T.-c. Fu, "A Review on Time Series Data Mining," *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 1, pp. 164-181, Feb. 2011.

[3] Y.-S. Moon and B. S. Lee, "Safe MBR-Transformation in Similar Sequence Matching," *Information Sciences*, Vol. 270, pp. 28-40, June 2014.

[4] X. Zhou, X. Zhou, L. Chen, and A. Bouguettaya, "Efficient Subsequence Matching over Large Video Databases," *The VLDB Journal*, Vol. 21, No. 4, pp. 489-508, Aug. 2012.

[5] Y.-S. Moon, K.-Y. Whang, and W.-K. Loh, "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, Apr. 2001.

[6] Y. Peng, R. C. Wong, L. Ye, P. S. Yu, "Attribute-Based Subsequence Matching and Mining," In *Proc. of the 28th Int'l Conf. on Data Engineering (ICDE)*, IEEE, Washington DC, pp. 989-1000, Apr. 2012.

[7] Y.-S. Moon, K.-Y. Whang, and W.-S. Han, "GeneralMatch: A Subsequence Matching Method in Time-Series Databases Based on GeneralizedWindows," In *Proc. Intl Conf. on Management of Data*, ACM SIGMOD, Madison, Wisconsin, pp. 382-393, June 2002.

[8] H.-S. Kim, M. Lee, and Y.-S. Moon, "A Dual Approach of GeneralMatch in Time-Series Subsequence Matching," In *Proc. of the 5th FTRA Intl Conf. on Computer Science and its Applications*, Danang, Vietnam, pp. 167-172, Dec. 2013.