

허밍 기반 음원 검색을 위한 오디오 특징 시퀀스 데이터 색인 기법 개발

송재중, 임태범
전자부품연구원

jcsong@keti.re.kr, tblim@keti.re.kr

Development of Audio Feature Sequence Data Indexing Method for Query

by Singing and Humming

Chai-Jong Song, Tea-Buem Lim
KETI

요 약

본 논문에서는 허밍기반 음원 검색 시스템을 위한 오디오 특징 시퀀스 데이터 색인 기법을 제안한다. 우선 Query-by-Singing/Humming (QbSH) 시스템의 특징 데이터베이스를 생성하기 위하여 MP3 와 같은 다성음원에서 주요 멜로디를 추출하여 시퀀스데이터를 생성하고, 고속 검색을 지원하기 위한 시퀀스데이터를 색인화한다. 본 논문에서는 최소 Dynamic Time Warping (DTW) 거리 기법, 시퀀스 추상화 기법, 상한 값 기반 DTW 기법과 같이 세 가지의 시퀀스 데이터의 색인화 기술을 제시하고 각각에 대한 문제점을 파악하고, 성능을 평가한다. 이를 통하여 향상된 검색 시간과 검색 정확도를 얻을 수 있다.

1. 서론

최근 음악이나 동영상과 같은 멀티미디어 데이터의 양이 폭발적으로 증가하고 다양한 기기를 이용해 이에 대한 사용이 일반화되고 있다. 이에 따라, 방대한 멀티미디어 데이터에 대한 효과적인 저장과 사용자가 필요한 데이터를 보다 정확하고 빠르게 검색하는 시스템의 중요성이 더욱더 커지고 있다. 이러한 요구에 맞추어 관련된 다양한 연구가 진행되어오고 있다.

음악 데이터 검색에 사용되는 방법은 크게 두 가지로 나누어 볼 수 있다. 첫 번째는 음악 데이터에 대한 메타데이터를 이용하는 방법이다. 어떤 음악에 대한 작곡자, 가수, 제목, 가사내용 등의 메타정보를 데이터베이스화 하고, 이들을 키워드로 사용자들이 질의를 내리면 SQL 인터페이스를 갖는 DBMS 가 검색을 수행하는 방법이다. 이 방법은 사용자가 적절한 키워드를 알고 있지 못한 경우에는 검색 자체가 불가능하다. 두 번째는 음악의 내용을 기반으로 검색하는 방법이다. 사용자가 정확한 키워드를 알고 있지 못하지만 어렵게 멜로디를 알고 있을 때 이 멜로디를 질의에 이용하게 된다. 즉, 사용자는 자신이 기억하는 멜로디를 허밍 하여 질의를 내리고, 허밍과 유사한 멜로디를 포함하는 음악데이터를 검색하는 방식이다.

허밍 기반 음원 검색은 허밍과 음악 데이터로부터 각각 음의 높낮이 (음조, pitch)와 음의 길이 (duration)를 특징데이터로 추출하여 이를 비교함으로써 가장 유사한 특징데이터를 포함하는 음원을 찾는 방식이다. 추출된 특징데이터는 시퀀스 데이터이고, 허밍기반의 질의 처리 과정은 시퀀스 데이터에 대한 유사성 검색의 문제로 변환된다.

시퀀스 데이터에 대한 유사성 검색에 가장 일반적으로 사용되는 유사성 측정 방법은 DTW 이다. 유클리디언 거리는 시퀀스 데이터를 구성하는 각 요소들을 독립적으로 다루기 때문에 길이가 다르거나 샘플링 비율이 다른 시퀀스 데이터간의 거리 측정에 적합하지 않았다. DTW 는 유클리디언 거리로 처리할 수 없는 문제를 해결하는 거리 측정 방식으로 시간 축에 대해 시퀀스 데이터를 확장하거나 축소 할 수 있도록 하여 데이터간의 최적 거리를 찾아낼 수 있다.

하지만, DTW 는 그 복잡도로 인해서 대용량 시퀀스 데이터베이스에서 사용하기에는 한계가 있다. 반면, 음악 데이터는 하루에도 전세계적으로 매우 많은 양이 새로 발생하며 이에 대한 허밍 검색을 지원하기 위해서는 검색 속도 측면의 고려가 필요하다. 본 논문에서는 DTW 기반의 거리측정 방법을 이용해서 검색 속도를 높일 수 있는 색인 방법을 제안한다.

2. 제안하는 시퀀스 데이터 색인 기술

음악 데이터에 대한 특징은 시퀀스데이터 형태로 추출되고 이를 위한 측정은 DTW 가 대표적이다. DTW 는 그 복잡도로 인해서 대용량 시퀀스 데이터베이스에서 사용하기에는 한계가 있다. 이러한 문제를 해결하기 위해서 다양한 색인 기법들에 대한 연구가 진행되어왔다 [1, 2, 3, 4, 5]. 이 방법들을 요약하면, 착오 기각 (False Negative)을 발생 시키지 않으면서 시퀀스 데이터베이스를 필터링 할 수 있는 방법을 이용하여 후보 집합을 찾아낸 후에 DTW 를 이용해 최종 결과를 찾아낸다. 기존에 제안된 필터링 방법들 중 가장 대표적인 것은 하한 거리 (Lower Bound)와 PAA (Piece-wise Aggregate

Approximation) 이다. PAA 는 시퀀스 데이터의 차원을 축소하는 방법이고 하한 거리는 질의 시퀀스 데이터에 상한 및 하한 범위를 부여하여 거리를 계산하는 방법이다. 하한 거리와 PAA 는 같이 사용하여 필터링 효과를 높일 수 있다.

[1]에서는 기존 방법과는 다르게 DTW 거리를 계산하는 도중에 주어진 값보다 큰 값을 구할 것이 예측될 때 계산을 멈추는 방법을 이용하여 속도를 높인다. [6]에서는 참조 시퀀스들을 선정하고 참조 시퀀스와 시퀀스 데이터간의 DTW 거리를 구해서, 유클리디언 공간상으로 시퀀스 데이터를 변환한다. 이를 통해 복잡도가 높은 DTW 대신 유클리디언 거리를 측정해서 검색을 수행하는 방법이다.

본 논문에서는 시퀀스 데이터 필터링에 적용할 수 있는 새로운 거리 측정 방식을 제안한다. 제안하는 색인 기법은 기존의 시퀀스 클러스터링 기법 등을 통해 유사한 시퀀스들을 하나의 그룹으로 묶고, 그룹을 대표하는 최소 하한 시퀀스 (MBS, Minimum Bounding Sequence)를 생성한다. 그리고, 질의 시퀀스와 각 그룹을 대표하는 MBS 들과의 최소 DTW 거리를 측정하여 질의와 유사한 시퀀스를 포함할 가능성이 있는 그룹을 찾아내고 그룹내의 시퀀스와의 DTW 거리 측정을 통해서 가장 유사한 시퀀스를 찾아낸다.

본 논문에서 제안하는 것은 유사한 그룹을 대표하는 Minimum Bounding Sequence (MBS) 와 MBS 와 시퀀스 간의 최소 DTW 거리 측정 방식이다. 또한, 이를 이용한 색인 구축 방법 및 질의 처리 방식을 제안한다.

최소 DTW 거리는 대용량의 시퀀스 데이터로부터 적절한 클러스터링 방법을 통해 유사한 시퀀스 데이터 그룹을 생성하고 이를 기반으로 DTW 거리를 측정한다. 시퀀스 클러스터링 방법 K-means 알고리즘을 사용한다. 생성한 시퀀스 그룹과 질의 시퀀스 사이의 유사도를 측정할 수 있다면 가장 가까운 시퀀스 그룹내의 시퀀스 데이터와 질의 시퀀스를 우선적으로 비교하여 유사한 시퀀스를 빠르게 검색 할 수 있다. 시퀀스 그룹과의 거리를 측정하기 위해서 시퀀스 그룹에 대한 Minimum Bounding Sequence (MBS)를 정의한다. MBS 는 그룹 내 시퀀스 데이터들에 대해서 각 차원의 최소 값과 최대 값을 구한 것이다. 그림 1 에서 위쪽은 유사한 시퀀스들을 모아 놓은 시퀀스 그룹이고, 아래쪽은 유사한 시퀀스들에 대한 MBS 이다. MBS 는 식 (1) 과 같이 정의 된다.

$$\begin{aligned}
 SG &= \{S_0, S_1, \dots, S_{n-2}, S_{n-1}\} \\
 S_i &= \{s_{i,0}, s_{i,1}, \dots, s_{i,n-1}\} \\
 MBS(SG) &= \{mbs_0, mbs_1, \dots, mbs_{n-1}\} \\
 mbs_i &= (l_i, u_i), \\
 \text{where } l_i &= \min(s_{0,i}, s_{1,i}, \dots, s_{n-1,i}) \text{ and} \\
 u_i &= \max(s_{0,i}, s_{1,i}, \dots, s_{n-1,i})
 \end{aligned}
 \tag{1}$$

MBS(SG) 와 질의 시퀀스 Q 와의 최소 DTW 거리는 식 (2) 에 의해서 계산된다. Q 와 MBS 의 I 번째 요소인 qi 와 mbsi 간의 최소 거리인 MinDist_i 를 먼저 정의한다. 최소 DTW 거리인 MinDTWDist 는 기본적으로 DTW 와 동일하게 계산하며 단지, 질의 시퀀스 Q 와 MBS 의 각 요소간 거리를 MinDTWDist 형태로 계산한다는 것이 다르다.

최소 DTW 거리를 시퀀스 그룹들을 필터링 하는데 사용하기 위해서는 어떤 시퀀스 그룹과의 거리는 그룹내의 모든 시퀀스 데이터들과의 거리보다 항상 작거나 같아야 한다.

$$\begin{aligned}
 Q &= \{q_0, q_1, \dots, q_{l-1}\} \\
 MinDist_i(q_i, mbs_i) &= \begin{cases} q_i - u_i, & \text{if } q_i > u_i \\ |q_i - l_i|, & \text{if } q_i < l_i \\ 0, & \text{otherwise} \end{cases}
 \end{aligned}
 \tag{2}$$

$$\begin{aligned}
 MinDTWDist(MBS(SG), Q) &= \\
 &MinDist_i + \\
 &\min \begin{cases} MinDTWDist(Rest(MBS(SG)), Q) \\ MinDTWDist(MBS(SG), Rest(Q)) \\ MinDTWDist(Rest(MBS(SG)), Rest(Q)) \end{cases}
 \end{aligned}$$

그림 1 에서 MinDTWDist 를 계산하는 예를 보여주고 있다. 총 6 개의 시퀀스 (S1, S2, S3, S4, S5, S6) 가 있으며 이들은 두 개의 시퀀스 그룹 SG1 와 SG2 으로 나뉘어져 있다. SG1 에는 (S1, S2, S3), SG2 에는 (S4, S5, S6) 이 포함되어 있다. SG1 과 SG2 에 대한 MBS(SG1) 과 MBS(SG2)가 각각 구해져 있다. 매트릭스는 MBS(SG1) 과 질의 시퀀스 Q 와의 MinDTWDist 를 구하는 것을 보여주고 있고, 아래쪽 매트릭스는 MBS(SG2) 와 질의 시퀀스 사이의 거리를 구하고 있다. MinDTWDist(Q, MBS(SG1))은 0 이고, MinDTWDist(Q, MBS(SG2))은 31 이다. 이로 보아 Q 와 보다 유사한 시퀀스들은 SG1 에 있으며 Q 와 S1, S2, S3 사이의 DTW 거리가 모두 31 보다 작다면 Q 와 가장 가까운 시퀀스는 SG1 에 있는 것을 확정할 수 있다.

SG1										SG2									
1	2	5	4	3	0	1	4	3	S1	3	6	8	9	10	9	6	7	6	S4
1	3	6	4	2	1	2	4	3	S2	4	6	9	9	11	7	7	7	7	S5
2	4	4	5	4	3	3	3	1	S3	5	5	6	7	7	6	6	9	9	S6
MBS										MBS									
1	2	4	4	2	0	1	3	1		3	5	6	7	7	6	6	7	6	
2	4	6	5	4	3	3	4	3		5	6	9	9	11	9	7	9	9	
Q										Q									
2	0	0	2	2	0	0	1	0		2	1	3	4	5	5	4	4	5	4
3	1	0	1	1	0	0	0	0		3	0	2	3	4	4	3	3	4	3
1	0	1	3	3	1	0	0	2		1	2	4	5	6	6	5	5	6	2
0	1	2	4	4	2	0	1	3		0	3	5	6	7	7	6	6	7	6
3	1	0	1	1	0	0	0	0		3	0	2	3	4	4	3	3	4	3
5	3	1	0	0	1	2	2	1		5	0	0	1	2	2	1	1	2	1
5	3	1	0	0	1	2	2	1		5	0	0	1	2	2	1	1	2	1
4	2	0	0	0	0	1	1	0		4	0	1	2	3	3	2	2	3	2
2	0	0	2	2	0	0	0	1		2	1	3	4	5	5	4	4	5	4
	1	2	4	4	2	0	1	3			3	5	6	7	7	6	6	7	6
	2	4	6	5	4	3	3	4			5	6	9	9	11	9	7	9	9
									MBS(SG1)										MBS(SG2)

그림 1. 시퀀스 그룹의 MBS 와 MinDTWDist 계산의 예

음원 특징의 데이터 시퀀스는 이웃하는 두 데이터의 값의 차이가 매우 큰 편이다. 이런 이유로 유사한 데이터 시퀀스들의 MBS 의 각 차원의 값이 데이터 집합이 가질 수 있는 최대 및 최소 값으로 나타난다. 따라서, 대부분의 클러스터의 MBS 가 전체 데이터 영역을 포함하게 되어 MBS 간의 변별력이 떨어지게 된다. 이 문제를 해결하기 위해서 도입한 방법은 시퀀스 추상화 Sequence Abstraction (SA) 기법이다.

시퀀스 추상화 기법이란 DTW 의 계산 비용을 줄이기 위해서 데이터 시퀀스 및 질의 시퀀스를 일정 간격단위로 추상화하여 시퀀스 자체의 길이를 줄이는 방법이다.

그림 2 에서 시퀀스 추상화의 예를 보여주고 있다. 시퀀스가 존재하는 영역을 일정 구간으로 나누고, 각 구간에 들어오는 서브 시퀀스의 상한 값과 하한 값을 해당 구간의 추상화 값으로 결정한다. 그림에서 첫 번째 구간의 추상화 값은 0~4 이고, 두 번째 구간의 추상화 값은 0~3 이 된다. 시퀀스의 차원이 100 이라고 가정하면 100 개의 값이 4 개의 (u : upper, l : lower)

쌍으로 줄어들게 되고 이에 상응하는 계산 비용 절감을 얻을 수 있다. 질의 시퀀스 역시 같은 방식으로 추상화하고 추상화된 두 시퀀스간의 거리는 MinDTWDist 를 이용해서 계산한다.

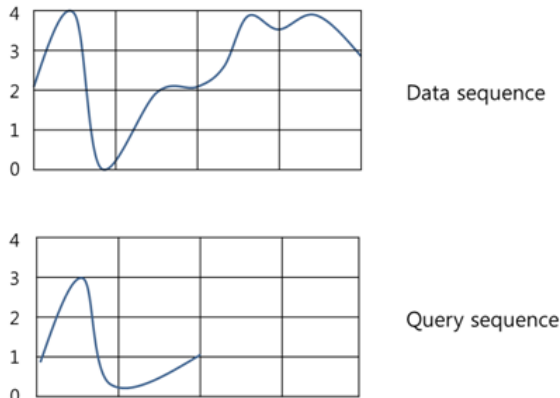


그림 2. 시퀀스 추상화 기법

그림 2 에서 보면 데이터 시퀀스는 $\{(0,4), (0,3), (2,4), (2,4)\}$ 로 추상화 될 수 있으며, 질의 시퀀스는 $\{(0,3), (0,0)\}$ 으로 추상화 된다. 추상화된 두 시퀀스간의 MinDTWDist 는 수식 (3) 과 같이 계산된다. 기본적인 개념은 MinDTWDist 와 동일하며 상한 및 하한 쌍 간의 MinDTWDist 를 계산한다는 것이 다르다. 수식에서 a 는 상수 값으로 추상화를 수행할 때 각 구간에 포함되는 차원의 수를 의미한다.

$$\begin{aligned}
 Q &= \{(q_{u_0}, q_{l_0}), (q_{u_1}, q_{l_1}), \dots, (q_{u_{i-1}}, q_{l_{i-1}})\} \\
 D &= \{(d_{u_0}, d_{l_0}), (d_{u_1}, d_{l_1}), \dots, (d_{u_{i-1}}, d_{l_{i-1}})\} \\
 \text{MinDist}((q_{u_i}, q_{l_i}), (d_{u_i}, d_{l_i})) &= \begin{cases} a \cdot |q_{u_i} - d_{l_i}|, & \text{if } q_{u_i} < d_{l_i} \\ a \cdot |d_{u_i} - q_{l_i}|, & \text{if } d_{u_i} < q_{l_i} \\ 0, & \text{otherwise} \end{cases} \quad (3) \\
 \text{MinDTWDist}(D, Q) &= \text{MinDist}_i + \min \begin{pmatrix} \text{MinDTWDist}(\text{Rest}(D), Q) \\ \text{MinDTWDist}(D, \text{Rest}(Q)) \\ \text{MinDTWDist}(\text{Rest}(D), \text{Rest}(Q)) \end{pmatrix}
 \end{aligned}$$

MinDTWDist 는 실제 DTW 와는 다르며 MinDTWDist 값으로는 질의 시퀀스와 실제로 유사한 데이터 시퀀스를 찾을 수 없다. 그럼에도 MinDTWDist 를 계산하는 이유는 MinDTWDist 를 이용해서 실제 DTW 계산을 수행할 후보집합을 찾아낼 수 있기 때문이다. MinDTWDist 가 후보집합을 선출할 수 있기 위해서는 다음과 같은 것을 만족해야 한다.

$$\text{MinDTWDist}(Q, D) < DTW(Q, D) \quad (4)$$

시퀀스 추상화는 계산비용을 대폭 줄인 추상화된 시퀀스 간의 계산 결과로 잠정적인 k-dist 를 구한 후 이를 이용해서 후보집합을 선출하고 후보 집합에 대해서 실제 DTW 계산을 수행하는 것이 핵심이다. 구간의 크기를 r 이라고 하고 데이터 시퀀스의 크기가 M 이고 질의 시퀀스의 크기가 N 이라면 계산 비용은 $O(M \cdot N)$ 으로 줄어들게 된다. 즉, 계산비용이 만큼 감소하게 된다.

MP3 로부터 추출한 데이터 시퀀스는 MIDI 와는 다르게 음성이 분리가 안 되므로 음성 및 악기 소리로부터 멜로디를 추출하므로 잡음이 섞이게 된다. 잡음은 이웃하는 데이터와의 차이가 매우 큰 값을 가지게 된다. 또한, 정확도를 높이기 위해서 크로마(Chroma)라는 전처리 과정을 수행하는데 이 과정 역시 이웃하는 데이터들 간의 차이를 크게 하는 효과가 있다.

실제로 실험을 통해서 살펴본 결과 데이터 시퀀스들과 질의

시퀀스간의 MinDTWDist 의 최대값이 DTW 의 최소 값보다 작은 경우를 많이 볼 수 있었다. 이것은 모든 데이터 시퀀스들이 후보집합으로 선정된다는 뜻으로 MinDTWDist 를 수행하는 비용이 DTW 를 수행하는 비용에 고스란히 더해 지게 된다.

이를 보완하기 위해서 구간을 값의 분포에 따라서 동적 구간을 이용한 시퀀스 추상화 기법인 Dynamic Sequence Abstraction (DSA) 을 제안한다. 그림 3 은 기본 구간의 크기를 4 로 하고 한 구간의 상한 및 하한 값의 차이를 2 보다 작게 하면서 추상화를 수행한 결과 이다. 이렇게 하더라도 구간 3~4 에서는 상한 값과 하한 값의 차이가 4 가 되는데, 이것은 바로 이웃하는 데이터간의 차이가 4 이므로 어쩔 수 없는 상황이다.

이때 수식 (3)에서 상수 a 는 비교하는 두 구간의 크기 중 최소값으로 한다. 즉, 그림 3 에서 데이터 시퀀스의 두번째 구간과 질의 시퀀스의 3 번째 구간을 비교한다면 a 는 구간의 크기가 작은 데이터 시퀀스의 두번째 구간의 크기인 1 로 한다. 이렇게 하면 구간이 상대적으로 많아져서 계산비용이 커지지만 구간의 상한 및 하한 값의 차이가 커지는 것을 완화할 수 있어서 MinDTWDist 를 통해서 선정하는 후보집합의 수를 줄일 수 있다.

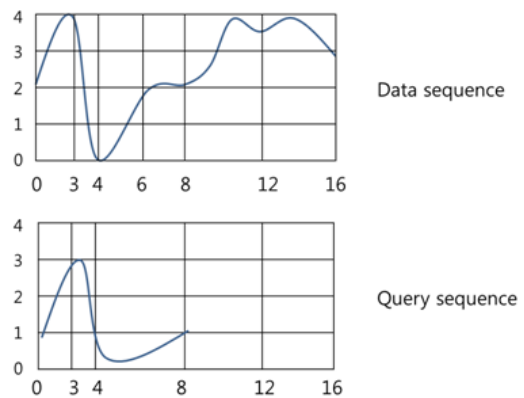


그림 3. 동적 구간 시퀀스 추상화 (DSA) 기법

위에서 제시한 두 접근 방법 모두 k 번째 데이터 시퀀스의 실제 DTW 거리(k-distance)를 이용해서 k-distance 보다 minDTWDist 가 더 큰 클러스터나 시퀀스에 대해서는 DTW 계산을 수행하지 않는 방법을 사용하고 있다. 이 경우에서처럼 k-distance 를 알고 있는 경우에는 DTW 수행 시 warping path 의 중간 누적 값이 k-distance 보다 커지면 그 다음 계산은 더 이상 할 필요가 없다. k 는 전체 데이터 시퀀스 수보다 훨씬 작기 때문에 대부분의 DTW 계산의 경우에 중간 누적 값이 k-distance 보다 커지게 될 것이다. 이렇게 중간에 계산을 멈추게 할 수 있다면 계산비용을 상당히 줄일 수 있는 상한 값 기반 Upper Bound DTW, UBDTW (UBDTW) 알고리즘을 제안한다. UBDTW 에서는 다음 단계에서 계산하지 않아도 되는 영역을 설정하고 이 영역 안에서만 warping path 계산을 수행하도록 제한한다. 허밍 시퀀스를 기준으로 음원 데이터 시퀀스들과 각각 거리를 계산하고 warping path 를 계산하는 것이 UBDTW 의 한 단계이다. 한 단계씩 진행이 될 때 마다 다음 단계에서 계산을 수행할 영역을 설정하는 것이다.

3. 실험 및 성능평가

본 논문에서는 제안하는 색인 기법을 검증하기 위해서 기존의 PAA 및 하한 거리를 이용한 색인 기법과 실험을 통해 비교하였다. 성능평가 척도로는 불필요한 DTW 연산의 수행횟수를

선정하였으며, 검색과정에서 수행되는 DTW 연산 횟수를 측정하였다. 실험에 사용된 데이터는 사람이 특정 노래의 일부분을 노래한 것으로부터 특징을 추출한 시퀀스 데이터이다. 총 200 개의 시퀀스 데이터를 실험에 사용하였다. 각 시퀀스 데이터의 차원은 750 이었다. 성능 비교를 위하여 제안하는 방법과 PAA 및 하한거리를 이용한 색인 방법을 Mac OS 10.7.2, gcc 4.1.1 을 기반으로 구현하였다. 이 실험에서는 별도의 클러스터링 방법을 사용하지 않았고 단순히 같은 곡에 대해서 허밍 한 노래는 같은 그룹으로 하였다. 질의는 총 1204 개의 서로 다른 질의를 수행하였으며, 1024 개의 질의 중 198 개는 시퀀스 데이터베이스에 저장되어 있는 시퀀스를 이용하였다. 질의의 형태는 K 의 값은 2, 5, 8, 11, 20 으로 바뀌가며 실험하였다. K 의 값을 변경할 때 마다 1024 개의 질의를 수행하였으며 이때 평균 DTW 연산 횟수를 측정하였다.

그림 4 은 실험 결과를 보여주고 있다. 그림에서 보는 것처럼 K 가 커질수록 제안하는 방법과 기존 방법의 DTW 계산 횟수 차이가 커짐을 알 수 있다. 제안하는 방법의 경우 그룹이 20 개 일때는 항상 20 회의 최소 DTW 거리를 계산해야 하며 이는 DTW 계산 횟수에 포함되었다.

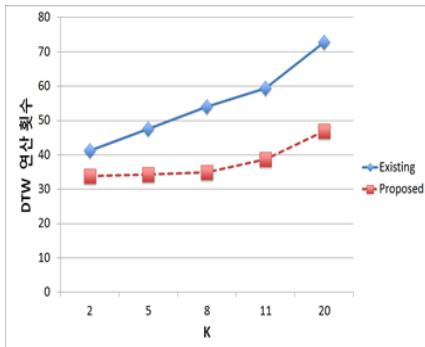


그림 4. 기존 PAA 기법과 제안하는 MBS 기법의 성능비교

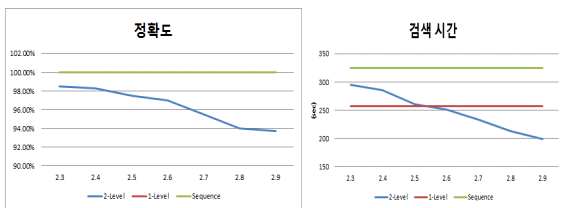


그림 5. 제안하는 기법의 검색 정확도 및 검색시간 실험

성능 및 정확도를 측정하기 위해서 2000 개의 데이터 시퀀스와 20 개의 질의 시퀀스를 가지고 실험을 수행하였다. HW 는 AMD 2.3GH 2 Core, 4GB 이었고, OS 는 Ubuntu 11 이었다. 그림 5 은 20 개 질의를 수행한 후 평균을 낸 결과를 비교한 것이다. level-2 는 구간의 크기를 4 일 때 와 2 일 때 색인을 생성한 것을 말하고, level-1 은 구간의 크기를 4 일 때만 색인을 생성한 것을 말한다. level-2 에서 a 값을 2.2 ~ 2.6 으로 변경하면서 결과를 측정하였다. level-1 에서는 a 값을 1.0 인 경우에 대해서만 측정하였다. a 가 1.0 인 경우에는 정확도가 100%를 보장한다. a 값을 2.2 에서 증가시킬수록 검색 성능이 향상되는 것을 볼 수 있다. 반면, 정확도는 조금씩 떨어짐을 볼 수 있다. a 값이 2.6 보다 작을 때는 오히려 level-2 의 색인들이 검색 성능을 떨어뜨리는 것을 알 수 있다. 이유는 구간의 크기를 각각 4 와 2 로 해서 색인을 구축했다고 하더라도 동적으로 구간을 결정하므로 실제 색인을 탐색하는 시간이 1/16 이나 1/4 로 줄어든

는 것이 아니라 각각 1/8, 1/2 정도로 줄어든다. 이로 인해서 정확도를 100% 가까이 유지하면서 검색성능을 높일 때는 색인을 하나만 이용하는 것이 오히려 효율적이다.

4. 결론

개발하는 음악 데이터 관리시스템에는 DTW 기반의 유사도 검색 성능을 높이기 위한 색인 기술을 새롭게 제안하여 구현하였다. 색인 기술은 시퀀스 그룹에 대한 MBS 를 정의하고, MBS 와 질의 시퀀스와의 거리를 계산하기 위한 최소 DTW 거리 기법을 포함한다. 또한, MBS 와 최소 DTW 거리를 이용하여 K-NN 질의 및 범위 질의를 처리하는 방법을 제안하였다. MP3 로부터 추출한 데이터 시퀀스들 및 메타데이터를 저장 및 검색하기 위한 색인 기법을 제안하여 구현하였다. 데이터 시퀀스와 질의 시퀀스간의 유사도를 비교하여 내용기반 검색을 지원할 수 있도록 한다. 이때 검색 성능을 향상시키기 위한 색인을 설계하고 구현하였다. 100%의 정확도를 유지할 때 약 21%의 검색시간을 줄일 수 있었다. 또한, 색인의 파라미터인 a 를 변화시키기에 따라 정확도와 검색성능에 변화가 있는데, 약 90%의 검색 정확도를 유지하면서 검색 시간을 50% 이상 향상시킬 수 있다.

참고문헌

- [1] J. Abfal, H.-P. Kriegel, P. Kröger, P. Kunath, A. Pryakhin, and M. Renz. "Similarity search on time series based on threshold queries". In EDBT, 2006.
- [2] Y. Chen, M. A. Nascimento, B. C. Ooi, and A. K. H. Tung. "SpADE: On Shape-based Pattern Detection in Streaming Time Series". In ICDE, 2007.
- [3] I. Assent, M. Wichterich, R. Krieger, H. Kremer and T. Seidl, "Anticipatory DTW for Efficient Similarity Search in Time Series Databases," Proceedings of the VLDB Endowment, pp. 826-837, 2009.
- [4] E. Keogh and C.A. Ratanamahatana, "Exact Indexing of Dynamic Time Warping," Knowledge and Information Systems, vol. 7, no. 3, pp. 358-386, 2005.
- [5] Y. Zhu and D. Shasha, "Warping Indexes with Envelope Transforms for Query by Humming," Proceedings of the ACM SIGMOD, pp. , 2003.
- [6] S. Salvador and P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Spaces," Proceedings of KDD Workshop on Mining Temporal and Sequential Data, pp. 70-80, 2004.