

## 음파 분석을 이용한 사용자 적응형 음악 추천 시스템

김동문<sup>o</sup>, 이지형

성균관대학교 전자전기컴퓨터 공학부

dmkim<sup>o</sup>@skku.ac.kr, jihlee@ece.skku.ac.kr

### User adapted music recommendation System using sound wave

Dong-Moon Kim<sup>o</sup>, Jee-Hyong Lee

Dept. of Electrical and Computer Engineering, Sungkyunkwan University

#### 요 약

최근 들어 음원 협회의 항소로 인해 음악 파일을 무료로 다운 받을 수 없게 되었다. 이로 인해, 유료 음악 사이트의 사용이 증가되었고, 수익성이 커지고 있다. 하지만 수요가 커진 것에 비해, 대부분의 음악 사이트들의 서비스는 음악 메일이나 휴대폰 전송 등에 그치고 있다. 따라서 사용자를 유치하기 위한 전략으로 추천시스템을 제안하고자 한다. 그 방법으로, 본 논문에서는 음악의 파형 변화를 분석하고, 사용자가 다운로드했던 파일의 리스트를 통하여 사용자 맞춤형 추천 시스템을 벡터 유사도를 통하여 구현하고자 한다. 음악에 대한 성분은 파형을 통하여 진폭과 진동수에 대한 특징 벡터를 추출한다. 그리고 사용자의 다운로드 리스트에 누적시킨다. 위의 두 절차를 통해 사용자의 리스트를 분석하여 비슷한 성분의 음악을 검색한다. 실험을 위해 사용되는 음악 성분에 대한 내용은 수치적인 데이터를 기반하고 있기 때문에 자동화가 용이했고, 빠른 연산 시간과 유동적인 검색 범위를 가질 수 있었다.

#### 1. 서 론

MP3라는 고품질 오디오 압축 기술이 발명되고 인터넷이 발달함에 따라서 음악을 온라인으로 접하기가 유용해졌다. 이에 사람들은 합법적이거나 혹은 불법적인 경로로 손쉽게 온라인에서 음악을 다운로드 받을 수 있었다.

하지만 IFPI(International Federation of Phonographic Industry)의 거센 소송에 의하여 불법 음악 공유에 대한 법적 제재가 강화되고 있다. 2006년 4월 5일을 기점으로 전 세계에 걸쳐 수백 명이 2,633유로(한화 394만원)의 벌금형에 처해졌고, 미국을 제외한 18개 국가에서 5,500건의 소송이 이루어지고 있다.

이러한 법적 제재가 강화됨에 따라 음악 불법 복제 행위가 줄어들고 있고, 사용자들은 합법적인 서비스로 전환할 수 밖에 없게 되었다.

국내도 예외는 아니다. 음악 사이트의 이용 고객이 날로 증가하고 있다. 지난 달, 벅스 뮤직은 492만 명, 소리바다는 649만 명의 방문자 수를 기록했다. 유료 사용자는 맥스 MP3가 25만 명(동년 6월), 멜론은 65만 명(동년 3월)에 달했다. 하지만, 이처럼 수요가 증가한데 비하여, 국내의 음악 사이트가 제시하고 있는 서비스는 음악 메일과 가사 보기, 휴대폰 전송 그리고 시대별 인기곡 추천 정도의 수준에 그치고 있다.

이에 본 논문에서는 음악 추천 시스템을 제안하고자 한다. 그동안 음악 추천에 대한 논문들을 살펴보면, 사용

자를 군집화하여 기후 등을 통하여 그 날의 기분에 맞는 음악을 추천[1]해주는 연구 사례가 있었다. 이에 본 논문에서는 음파를 STFT 형식으로 분석하여 벡터 유사도(Vector Similarity)를 통해 음악을 정렬한다. 이는 연산 과정을 자동화하며 속도를 향상시키는 장점을 보인다.

본 논문의 구성은 다음과 같다. 제 2장에서는 추천 시스템을 가동하기에 앞서 필요한 관련 연구를 기술하고 제 3장에서는 추천 시스템을 구현한다. 제 4장에서는 문제점 및 해결안을 분석하고 제 5장에서는 실험 결과를 확인한 후 제 6장에서는 결론 및 향후 과제에 대해서 기술한다.

#### 2. 관련 연구

##### 2.1 음악 특징 추출

음악이란 파형으로 이루어져 있다. 파형의 특징 종류로는 음률의 높이(pitch), 소리의 세고 여림(loudness), 음의 길이(duration), 음질(timbre)등이 있고 좀더 상세적인 특징으로는 STFT(Short Time Fourier Transform)에 기반을 둔 ZCR(zero crossing-rate) Spectral centroid, Spectral roll-off, Spectral flux와 지각적인 특징을 표현하는 MFCC(Mel-Frequency Cepstral Coefficients), LPC(Linear Prediction reflection Coefficients) 등이 있다[2]. 본 논문에서는 ZCR, Spectral roll-off, Spectral flux를 조합하여 사용한다. 이들 조합은 일정 구간 내에서 음악을 찾는 알고리즘에서 높은 정확도를 보여준 바 있으므로 채택하게 되었다[2].

2.1.1 ZCR

ZCR은 주어진 구간 내에 음성 신호가 기준선인 0을 통과하는 횟수를 측정하는 것이다. 즉, 이산 신호에서 연속 샘플링 값이 서로 다른 부호일 때 발생하는데, 이는 음성의 분할, 분석, 인식에 매우 유용하게 쓰인다. 음성의 신호의 크기를  $s$ 라 하고 샘플  $n$ 에 대한  $i$ 차 ZCR  $Z_i$ 는 다음과 같이 정의된다[2].

$$Z_i = \sum_{n=1}^N \frac{|sgn|s_i(n)| - sgn|s_i(n-1)||}{2}$$

단,  $sgn|s_i(n)| = \begin{cases} 1 & s_i > 0(n) \\ -1 & s_i < 0 \end{cases}$   
 $s =$  신호의 크기,  $n =$  샘플,  $i =$  차수

2.1.2 Spectral Roll-off

Spectral roll-off는 음성 구간과 무 음성 구간 사이를 구분하는 특징으로써 centroid와 함께 스펙트럼 형태의 EH 다른 측정 방법이다. roll-off의 정의는 크기(magnitude) 분포의 85%가 집중해 있는 주파수  $R_t$  이하를 말한다. 즉, 신호의 에너지 중 낮은 신호에 얼마나 많이 집중되어 있는가를 보여준다[2].

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 * \sum_{n=1}^N M_t[n]$$

$M_t[n]$ 은 프레임  $t$ 와 주파수  $n$ 에서 Fourier Transform의 크기

2.1.3 Spectral Flux

Spectral Flux는 스펙트럼의 변화율을 측정하는 방법으로써 연속된 스펙트럼의 분포의 정규화 된 크기들의 차를 제곱한 것을 의미한다. Flux는 음악과 음성을 구분 짓는데 적합하다[2].

$$F_t = \sum_{n=1}^N (N_t[n] - N_{t-1}[n])^2$$

$N_t[n]$ ,  $N_{t-1}[n]$ 은 각각 현재 프레임  $t$ 와 이전의 프레임  $t-1$ 에서 Fourier transform의 정규화된 크기

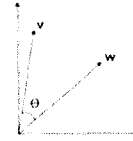
2.2 벡터 유사도 소개

벡터란 방향과 크기를 가지는 양의 단위이다. 벡터  $v$ 와  $w$ 가 존재할 때, 임의의 하나의 벡터를 평행 이동 시키면, 벡터들은 서로 간의 각을 이루게 된다. 벡터의 내적을 구하는 두 가지 공식을 이용하게 되면 두 벡터 간의 Cosine Distance를 얻게 되는데 이는 1에 가까운 값을 가지게 될수록 각은 좁아지게 된다.

$$|v||w|\cos\theta = v \cdot w$$

$$\cos\theta = \frac{v \cdot w}{|v||w|}$$

단,  $v \cdot w = (x_v \times x_w + y_v \times y_w + \dots + z_v \times z_w)$



<그림 1> 벡터  $v$ 와  $w$ 가 이루는  $\cos\theta$  연산

3. 제안된 추천 시스템 방법

음악에 대한 성향 벡터는 ZCR, Roll-off, Flux를 3개의 벡터 값으로 채택한다. 사용자의 성향을 추측하기 위해서는 한 번 이상의 다운로드 횟수가 필요하다. 사용자가 다운로드 했던 음악의 성향을 누적하여 평균값을 구한다.

$$\left( \sum_{i=1}^n ZCR_i, \sum_{i=1}^n Roll-off_i, \sum_{i=1}^n Flux_i \right) \div n$$

$n =$  다운로드 횟수

구해진 사용자의 성향 벡터와 노래의 성향 벡터를 내적 공식에 의하여 Cosine Distance값을 구할 수 있다.

$$Dc(h_{user}, h_{song}) = \frac{\sum_{i=1}^B h_{user}^i h_{song}^i}{\sqrt{\sum_{i=1}^B (h_{user}^i)^2 \times \sum_{i=1}^B (h_{song}^i)^2}}$$

위에서 구해진 Cosine Distance의 값은 0에서 1사이의 값을 가지게 된다. 이 때 Cosine Distance의 값이 1과 근접할수록 벡터간의 각도는 0°에 가깝게 되어 이들은 유사한 값을 가지게 된다. 따라서 Cosine Distance의 값을 역순으로 정렬하면 사용자가 다운로드 했던 음악과 유사한 성향의 음악을 순차적으로 표현할 수 있다.

4. 문제점 및 해결안

4.1 문제점

성향을 수치적으로 분석하여 유사도를 분석함으로써, 보다 객관적이고 신뢰감 있는 분석이 가능하게 되었다. 하지만 음악 파일의 숫자가 증가할수록 연산 시간은 그에 비례하여 증가하게 된다. 현재의 음악 사이트들은 매일 수백에서 수천여 곡에 음악이 업데이트되고 있다. 사용자의 성향과 전체 곡의 성향을 모두 비교하는 것은 시간이 오래 걸릴 뿐만 아니라 전체 시스템에 무리를 주게 된다.

4.2 해결안

음악에 대하여 분석된 성향 값을 임의의 기준 벡터를 선정하고, 기준 벡터와 음악 성향 벡터와의 벡터 유사도

를 미리 연산한다.

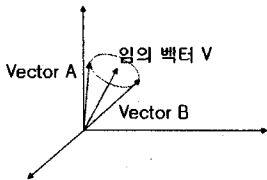
음악 파일들에 미리 연산된 값을 입력해놓게 된다면, 사용자의 성향 벡터를 기준 벡터와 이루는 벡터 유사도를 구하여 그 크기를 비교한 다음, 해당 벡터들과 연산한다면, 벡터 유사도 연산 횟수를 줄일 수 있게 된다.

이 때 또다른 문제점이 발생하게 된다. cos함수의 그래프는 0°를 기준으로 대칭형 분포를 이루고 있다. 그러므로 기준 벡터와 일정각  $\theta^\circ$ 를 이루는 벡터와  $-\theta^\circ$ 를 이루는 cos값은 같게 된다. 이는 기준 벡터와 만들 수 있는 각이 늘어날수록 오차에 해당하는 값이 늘어날 수 있다. 즉, 벡터가 생성되는 차원의 수가 증가함에 따라 오차값도 증가한다.

본 논문에서는 ZCR, Roll-off, Flux로 벡터를 표현했을 때 매칭률이 높았다[2]는 기록을 토대로 하고 있으므로, 3개 데이터를 토대로 3차원 벡터로 표현하여 연산된 값에 대한 오차와 그 범위에 대하여 논하고자 한다.

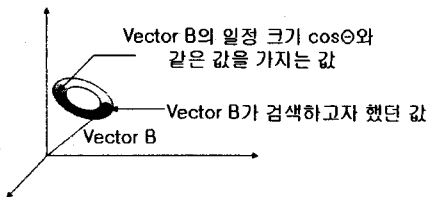
#### 4.3 3차원 벡터에서의 오차의 범위 및 정확도

벡터는 크기와 방향을 가지는 척도이지만, 방향에 중점을 둔 이론을 토대로 하고 있으므로, 크기에 해당하는 값은 비중을 두지 않도록 한다. 이 절에서는 크기를 기준 벡터와 같다고 한다. 이는 4.2절에서 논한 바와 같이  $\theta^\circ$ 의 값이 같아지는 3차원 벡터로 표현하면 다음 <그림 2>과 같다.



<그림 2> 같은 값을 가지는 3차원 벡터

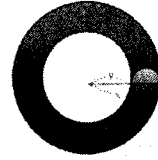
Vector A와 Vector B를 연결하는 원형에 위치한 벡터들은 모두 같은 값을 가지게 된다. Vector B를 기점으로 일정 크기의 cos distance안의 성향 벡터들을 분석하게 되면 같은 값을 가지는 벡터들은 <그림 3>와 같다.



<그림 3> Vector B와 같은 크기를 가지는 성향 벡터

그림 2에서 검색된 값의 안쪽 반지름을 y라 하고 검색하고자 했던 원의 반지름을 x라 하자. 이를 그림으로 나

타내면 다음과 같다.



<그림 4> 검색된 데이터의 정확도

찾고자 했던 데이터의 값은 원의 면적 공식을 이용하여 다음 식으로 표현할 수 있다.

$$\pi \left(\frac{x-y}{2}\right)^2$$

그리고 검색된 전체 데이터는 다음 식으로 표현할 수 있다.

$$\pi x^2 - \pi y^2$$

따라서 찾고자 했던 데이터의 정확도를 확인하기 위하여 검색된 데이터의 값으로 나누어 보면 다음 식과 같다.

$$\frac{\pi \left(\frac{x-y}{2}\right)^2}{\pi x^2 - \pi y^2} = \frac{\pi (x-y)^2}{4\pi (x-y)(x+y)} = \frac{x-y}{4(x+y)}$$

위 식에서 확인할 수 있듯이 찾고자 하는 기준 벡터에서 멀어질수록 x의 크기가 증가하기 때문에 그 정확도는 낮아진다.

기준 벡터를 찾고자 하는 영역이 포함하는 경우, 즉  $y < 0$ 이 되는 경우에는  $(x+y)$ 가 찾고자 하는 원의 지름이 되며, 이 때의 정확도는 아래와 같다.

$$\frac{(x+y)^2}{4x^2}$$

위의 식처럼 음악 전체가 아닌 일정 영역만큼의 검색만을 할 수 있으므로, 이는 성능을 향상시키는데 큰 도움이 될 수 있다.

#### 5. 실험결과

제한한 방법의 유용성을 보이기 위하여 추천 결과와 시간의 감축을 분석하였다. 추천 결과는 임의의 파일을 사용자 리스트에 선택 후 추천되는 음악의 성향을 파악한다. 시간의 감축은 기준 벡터의 존재로 인하여 어느 정도의 시간이 절약되는지 관찰한다.

실험 환경은 Windows XP 플랫폼에서 MATLAB을 이용하여 각 특징 벡터를 추출하였다. 음악 DB는 국내외의 음악 중 Ballad, Rock, R&B, Hip-hop, Classic에서 20곡씩 랜덤으로 추출하여 총 100개의 표본을 사용하였다. 음악 파일은 32bit 2channel 44100Hz 형식을 사용하였다. 실험을 위하여 추출된 값은 우선 음악의 길이에

<표 1> 추천 검색 결과

선택 음악 검색 순위	나 때문에 (Ballad, 여)	November (Rock, 남)	Straight Up (R&B, 여)	Without you (Hip-Hop, 남)	호두까기 인형 행진곡 (Classic)
1	해바라기 (Ballad, 남)	My immortal (Rock, 여)	Rumors (Hip-Hop, 여)	고해 (R&B, 남)	Trumpet Mardhing (Classic)
2	여자이니까 (Ballad, 여)	Pollution (Rock, 남)	Back at one (R&B, 여)	뉘시 (Hip-Hop, 남)	아베 마리아 (Classic)
3	그런일은 (Ballad, 여)	Beautiful People (Rock, 남)	내 곁에 있는 사람들을 위한 기도 (Hip-Hop, 남&여)	Hard to say I'm sorry (R&B, 남)	트라치 트라쉬 포카 몬레이커 (Classic)
4	소망 (Ballad, 여)	미뉴에트 (Classic)	고해 (R&B, 남)	Because (발라드, 여)	벳노래 (Classic)
5	She's all got it (R&B, 여)	She's gone (Rock, 남)	윈타임 (Hip-Hop, 남)	Rumors (Hip-Hop, 여)	사제중 가을 1악장 (Classic)
6	체념 (Ballad, 여)	Smells like teen spirit (Rock, 남)	Hard to say I'm sorry (R&B, 남)	Straight up (R&B, 여)	백조의 호수 (Classic)
7	CB Mass는 내 친구 (Hip-Hop, 남)	늘 (Ballad, 여)	뉘시 (Hip-Hop, 남)	웃음 속에 가려진 눈물 (발라드, 여)	즐거운 행진곡 (Classic)
8	Pretty fly (Rock, 남&여)	Waterfalls (R&B, 여)	Slow Jam (R&B, 남)	내 안의 그대 (Ballad, 여)	사랑의 슬픔 (Classic)
9	love론 (Hip-Hop, 남&여)	Hungarian Dance No.5 (Classic)	All my life (R&B, 여)	빈 수레 (Hip-Hop, 남)	Creep (Rock, 남)
10	Sukiyaki (Rock, 남)	Green day (Rock, 남)	Because (발라드, 여)	Rock is dead (Rock, 남)	파야노 소나타 21번 (Classic)

의하여 나누어진다. 그리고 각 벡터들의 평균값을 구한 후, 평균값으로 추출된 성향을 나누었다. 이를 통해 성향을 동등한 비율로 반영할 수 있도록 하였다.

5.1 추천된 결과

실험 결과 같은 장르 내에 음악이 주를 이루었고 비슷한 음색을 가진 다른 장르의 음악이 그 뒤를 이었다. 또한 같은 성별의 음악이 높은 순위를 나타냈다. 음악 분류 중 R&B와 Hip-Hop 음악의 경우는 다른 장르의 음악과 이루는 경계가 모호했다. 따라서 장르를 떠나 비슷한 선율이나 음악의 높낮이 쪽을 많이 반영하게 되었던 것으로 보인다. 특히 가사가 없고 악기에 의한 연주가 전 반부를 차지하는 Classic의 경우 10개의 추천 음악 중 9개가 Classic 음악을 추천해줌으로써 이는 음악의 성향이 잘 구분된 것이라고 판단된다.

5.2 시간의 감축

위에서 제안한 기준 벡터는 (1,1,1)로 사용했다. 이는 실험에 앞서서 벡터들을 가공할 때 평균값으로 나누었기 때문에 각 벡터들은 평균값을 1로 가지게 되기 때문이다. 정렬된 cos distance를 크기에 따라 분리해본 결과, 벡터들의 분산이 일정하지 않았기 때문에 사용자의 벡터에 따라 일정치 않은 개수의 표본 n이 나오게 되었다. 따라서 사용자의 벡터에 대하여 근접한 벡터의 일정한 개수의 표본 n을 추출하여 검색하였다. 이는 검색의 범위를 줄이는데도 도움이 되었고, n/total 만큼의 cos distance 비교 연산 시간 감축 효과를 가져 올 수 있었다.

6. 결론 및 향후 과제

음악 사이트의 개발 가치가 커지고 이용하려는 사용자가 유동적이다. 추천 시스템이 도입한다면 타사와의 경쟁에 유용하고 적합한 시기가 생각된다. 음원 분석을 통하여 자동화가 가능하므로 서비스를 유지하는 운영자의 부담을 덜 수 있다. 또한 연산이 신속하고 수치적인 데이터를 얻을 수 있다는 강점을 보였다. 하지만 추천에 사용된 값에서 각 성향에 대한 가중치를 적용하지 못하였다. 이 점에 대한 연구가 향후의 과제로 생각된다.

감사의 글 : 본 연구는 21세기 프론티어 연구개발 사업의 일환으로 추천되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크원천기반기술 개발사업의 지원을 받았습니다.

7. 참고 문헌

[1] 유지오, 퍼지 베이저안 네트워크와 효율성 이론을 사용한 상황 기반 음악 추천, 연세대학교, Ph.D thesis, Feb, 2006.  
 [2] G. Tzanetakis, "Manipulation, m Analysis and retrieval systems for audio", Princeton University, Ph.D thesis, June 2002.  
 [3] 차지혜, "다중 특징 벡터를 이용한 고속 오디오 검색", 한국해양대학교, Ph.D thesis, Feb 2002.