

사용자의 취향을 고려한 음악 재생 목록 생성 시스템

A Playlist Generation System based on Musical Preferences

방성우* · 김태연* · 정혜욱* · 이지형** · 김용세**

Sun-Woo Bang, Tae-Yeon Kim, Hye-Wuk Jung, Jee-Hyong Lee and Yong Se Kim

* 성균관대학교 전자전기컴퓨터공학과

** 성균관대학교 기계공학부

요 약

음악의 생산과 수요 증가와 함께 사용자의 장치에 저장되어 있는 음악을 관리하기 위한 관심 또한 증가하고 있다. 일반적으로 사용자는 음악을 효과적으로 관리하기 위해 재생 목록을 작성하고 이를 선택하는 방법을 사용하고 있다. 하지만 현재 사용되는 재생 목록의 작성 방법은 음악을 사용자가 직접 선택해야 하는 한계를 안고 있다. 따라서 재생 목록을 자동으로 작성하여 사용자에게 제공해주는 방법이 필요하다. 본 논문에서는 사용자의 음악사용의 상황과 취향을 고려하여 자동으로 재생 목록을 생성해주는 시스템을 제안한다. 이 시스템은 음악적 무드 (Musical mood) 분류 시스템과 음악 추천 시스템, 두 가지 별개의 시스템으로 구성되어 있다. 사용자는 음악을 추천 받기 위해 단지 하나의 음악을 선택한다. 그러면 시스템은 자동으로 재생 목록을 생성하기 위해 선택된 음악과 유사한 무드의 음악을 재생 목록에 추가한다. 사용자는 재생 목록에 추가된 음악 중 자신의 취향에 맞지 않는 음악을 제거하여 취향에 적합한 음악을 반복적으로 추천 받을 수 있다. 본 논문에서 제안하는 시스템의 실험과 평가를 위해 실제 음악을 수집하였으며 시스템을 통해 생성된 재생 목록을 분석하여 사용자의 취향이 보다 정확히 반영된 것을 확인하였다.

키워드 : 음악 추천, 음악 무드, 재생 목록 생성기, 지능 시스템, 내용 기반 분석.

Abstract

The rise of music resources has led to a parallel rise in the need to manage thousands of songs on user devices. So users are tend to build play-list for manage songs. However the manual selection of songs for creating play-list is bothersome task. This paper proposes an auto play-list recommendation system considering user's context of use and preference. This system has two separate systems: mood and emotion classification system and music recommendation system. Users need to choose just one seed song for reflection their context of use and preference. The system recommends songs before the current song ends in order to fill up user play-list. User also can remove unsatisfied songs from recommended song list to adapt user preferences of the system for the next recommendation process. The generated play-lists show well defined mood and emotion of music and provide songs that user preferences are reflected.

Key Words : Music recommendation, Musical mood and emotion, Play-list generator, Intelligent system, Content-based analysis.

1. 서 론

음악 추천 시스템은 음악의 생산과 수요의 증가와 함께 사용이 증가 되고 있다. 또한 온라인으로 음악을 제공하는 서비스의 사용이 빈번해 지면서 날로 증가하는 수많은 음악을 관리하기 위한 새로운 방법들이 요구되고 있다. 일반적으로 사용자들은 휴대용 저장장치나 개인 컴퓨터에 저장되

어 있는 음악들을 효과적으로 관리하고 다양한 상황에 사용하기 위해 다수의 재생 목록을 작성하고 사용자의 상황에 알맞은 재생 목록을 선택하는 방법을 선호하고 있다. 즉, 사용자는 듣고 싶은 음악을 여러 개의 재생 목록에 저장하고 사용자의 상황에 적합한 재생 목록을 선택하는 경향이 있다. 예를 들면, 사용자가 운동을 하고 있는 경우, 사용자는 여러 개의 재생 목록 중 비교적 음악이 빠르고 신나는 음악이 담겨 있는 재생 목록을 선택할 것이다. 만약 사용자가 도서관에서 공부를 하고 있을 경우에는 조용하고 집중력을 높여 줄 수 있는 비교적 차분한 음악이 저장된 재생 목록을 선호할 것이다. 하지만 사용자는 여러 가지 상황에 적합한 재생 목록을 작성하기 위해 저장 장치에 담겨있는 음악을 일일이 선택하여 직접 재생 목록에 저장해야 하는 번거로운 작업을 수행하여야 한다. 이런 방법은 시간이 많이 요구되고 자신이 원하는 음악을 판단하기 어려운 경우 재생 목록

접수일자 : 2010년 4월 3일

완료일자 : 2010년 5월 17일

본 논문은 본 학회 2010년도 춘계 학술대회에서 선정된 우수논문입니다.

+교신저자

감사의 글 : 본 연구는 지식경제부 산업원천기술개발 사업의 지원을 받았음.

작성의 문제를 가지고 있다. 따라서 재생 목록을 자동으로 작성하여 주고 사용자의 음악적 취향 또한 고려해 주는 음악 목록 생성 시스템이 필요하다. John C. Platt는 Gaussian process [1]를 학습하는 방법을 통해 사용자에게 자동으로 음악 재생 목록을 제공해 주는 시스템을 제안하였다. 이 방법은 다른 시스템과 비교하여 비교적 효과적인 음악 재생 목록을 제공하여 주지만 사용자에게 자동으로 재생 목록을 제공하여 주는 것에 중점을 두기 때문에 사용자의 음악 장르적 취향이나 사용의 상황을 고려하지 못한다. 사용자의 재생 목록에는 사용자가 듣고 싶은 음악이 저장되어 있어야 한다. 사용자들은 보통 여러 종류의 음악적 장르 취향을 가지고 있다. 따라서 재생 목록 생성 시스템은 음악 장르와 같이 사용자의 취향 또한 고려해 주어야 한다. 사용자의 음악적 취향을 고려하기 위한 다양한 방법 중 일반적으로 사용되는 방법은 내용 기반 (content-based) 음악 분석 방법이다. 이 방법은 음악의 신호만을 고려하여 음악을 분석하므로 음악 추천 과정에서 음악이 가지고 있는 메타데이터 없이 음악 분석이 가능하다. Deliège and Pedersen [2]은 내용 기반 음악 분석을 적용하여 fuzzy lists를 작성하고 이를 통해 재생 목록을 관리하고 음악을 추천하여 주는 방법을 제안하였다. 이 방법은 사용자의 재생 목록을 분석하여 사용자의 취향을 반영하는 음악을 추천하여 주기는 하지만 사용자의 음악사용 상황을 재생 목록에 적용하지 못하고 사용자의 재생 목록이 존재하지 않을 경우 재생 목록 생성이 불가능한 문제를 안고 있다.

본 논문에서는 사용자가 음악을 제공 받는 다양한 상황을 고려할 뿐만 아니라 사용자가 선호하는 음악을 자동으로 제공하여 주는 새로운 자동 재생 목록 생성 시스템을 제안한다. 사용자는 자신이 음악을 제공 받는 상황에 적합한 하나의 음악만을 선택함으로써 자동으로 음악 재생 목록을 제공 받을 수 있다. 이를 통해 사용자가 미리 재생 목록을 작성할 필요 없이 음악 재생 목록을 편리하게 제공 받을 수 있다. 이렇게 선택된 음악은 시스템 내에서 분석되어 사용자의 음악사용 상황에 적합한 음악들이 분류 및 추천되고 사용자는 재생 목록에 자동으로 저장된 음악을 즐긴다. 또한 사용자는 자신의 음악적 취향을 시스템에 적용하기 위해 듣고 싶지 않은 음악을 제거하여 다음 추천과정에서 자신의 음악적 취향이 반영되도록 할 수 있다. 이 과정의 반복을 통해 사용자는 점진적으로 자신의 취향이 보다 더 잘 반영된 음악을 추천 받을 수 있다. 본 논문에서 제안하는 시스템의 평가를 위해 실제 음악을 수집하였으며 시스템이 생성한 음악 재생 목록은 뛰어난 음악적 무드 분류 결과를 보였고 사용자의 취향에 적합한 음악을 제공하는 것을 확인하였다. 제안하는 시스템은 다음 4가지 사항을 고려하여 개발되었다.

1. 재생 목록이 없는 경우에서의 음악 추천
2. 사용자의 음악사용 상황을 고려 (음악적 무드)
3. 사용자의 취향을 고려한 음악 추천
4. 음악 재생 목록 자동 생성

본 논문에서는 관련 연구와 음악적 특징 추출에 관한 내용을 2장에서 설명하고, 3장에서 제안 시스템인 음악적 무드 분류 시스템과 추천 시스템을 소개한다. 4, 5 장에서는 제안 시스템의 평가를 위한 실험 설계 및 분석 결과를 보여준다. 마지막으로 결론을 통해 제안 시스템의 개선점과 향후 연구계획을 소개한다.

2. 기반 연구

음악의 신호를 내용 기반 방법으로 분석하기 위해 음악의 특징을 대표하는 요소를 추출하여야 한다. 각 음악 요소들은 음악 신호를 지정된 윈도우 크기로 분할하여 각 부분에서 값을 추출하고 추출된 값 전체의 평균값과 표준편차 값을 계산하여 얻어진다. 본 연구에서는 12개의 요소를 추출하였으며, 이 요소들은 음악의 특징을 비교하기 위해 일반적으로 추출되는 다수의 중요 요소 중 일부이다. 표 1은 추출된 음악적 특징 요소의 자세한 내용을 보여준다.

표 1. 음악적 특징 요소.
Table 1. Musical features.

특징 요소	요소 설명
Spectral Centroid	음악의 밝기 정도를 측정
Spectral Rolloff Point	주파수의 진폭변화를 측정
Spectral Flux	프레임 당 스펙트럼 모양의 변화 정도를 측정
Compactness	신호의 잡음 정도를 측정
Spectral Variability	신호 진폭 스펙트럼의 표준편차를 계산
Root Mean Square (RMS)	신호의 파워를 측정
Fraction of Low Energy Windows	다른 신호와 비교하여 신호의 조용한 정도를 측정
Zero Crossings	잡음의 양을 측정
Strongest Beat	신호에서 가장 비트가 큰 신호
Beat Sum	음악에서 사용된 비트의 합
Strength of Strongest Beat	다른 비트에 비해 가장 비트가 강한 비트의 세기를 측정
LPC	낮은 bit rate에서 신호의 질을 측정

이렇게 추출된 요소를 그림 1과 같이 Thayer의 음악 무드 모델 [3]에 적용하여 음악을 무드 별로 분류하였다. Thayer 모델은 음악의 무드를 Valence와 Arousal 축으로 이루어진 2차원 평면에 대입함으로써 음악의 무드를 보다 직관적으로 서술하기 위한 대표적인 측정 모델로 주로 사용된다. 가로축인 Arousal 축은 음악에서 “무언가를 하기 바쁜”으로 해석되는데 이는 음악의 비트나 음역의 변화정도를 측정하여 음악의 빠르기를 표현한다. 그리고 세로축인 Valence 축은 음악의 밝기나 크기 등을 의미하며 밝은 음악과 어두운 음악의 무드를 의미한다. 음악의 클래스를 Thayer 모델에 적용하여 이해하면 그 클래스 정의가 용이하고 합리적이기 때문에 본 논문에서는 음악 무드의 클래스를 이 모델에서 제안하는 4개의 무드 클래스 (A, B, C, D)로 정의한다. 음악의 무드는 서론에서 거론된 바와 같이 사용자의 환경에 적합한 음악을 추천하는데 활용한다. 각 클래스에 포함되는 음악 무드는 다음과 같다.

- A : aggressive, angry, fiery, hostile, malevolent, visceral, volatile
- B : lively, bright, humorous, happy, cheerful, exciting, joyous, playful
- C : sad, melancholy, gloomy, ominous
- D : dreamy, smooth, peaceful, plaintive, relaxed, sentimental

또한, 음악의 장르를 분류하기 위해 Mel-Frequency Cepstral Coefficients (MFCCs) [4]를 추출하였으며, 추출 정보를 이용하여 Hidden Markov Model을 생성하고 유사도 분석을 통해 음악의 장르를 비교 및 분류하였다.

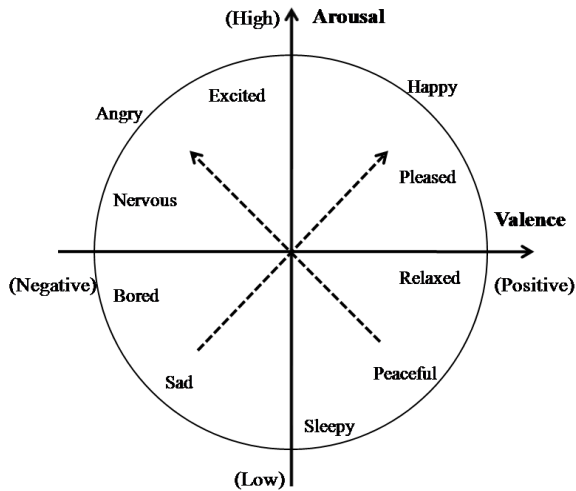


그림 1. Thayer의 음악 무드 모델.
Fig. 1. Thayer's model of musical mood.

3. 음악 추천 시스템

그림 2는 본 논문에서 제안하는 음악 추천 시스템의 개요를 보여주고 있다. 음악 추천 시스템은 음악 무드 분류 시스템과 추천 시스템, 두 가지 별개의 시스템으로 구성되어 있다. 먼저 음악 무드 분류 시스템에서는 사용자가 선택한 한 곡의 음악 무드를 정의하여 주고, 그 음악과 같은 무드의 음악들을 추천 시스템에 전달한다. 그리고 추천 시스템에서는 음악을 장르별로 분류하여 사용자의 음악 장르적 취향에 적합한 음악을 재생 목록에 추가한다. 그림 3에서 이 과정의 예를 추상적으로 표현하였다.

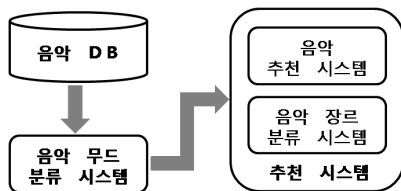


그림 2. 시스템 개요.
Fig 2. The diagram of system.

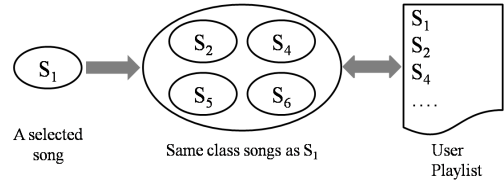


그림 3. 추천 과정.
Fig. 3. Recommendation process.

3.1 음악적 무드 (Musical mood) 분류 시스템

사용자의 상황에 적합한 음악을 추천하기 위해 음악의 무드, 무드에 기반을 두어 음악을 분류하는 시스템이 필요하다. 먼저 음악을 음악의 무드 별로 분류하기 위해 음악 신호에서 추출이 가능한 Spectral centroid, Spectral rolloff point, Spectral flux, Compactness, Spectral variability, Root mean square (RMS), Fraction of low energy windows, Zero crossing, Strongest beat, Beat sum, Strongest beat, Strength of strongest beat, LPC 요소를 추출 하였다. 그리고 이렇게 추출한 요소를 기반으로 k-NN (k-Nearest Neighbor) 분류 알고리즘 [5]을 사용하였다. k-NN 분류 알고리즘은 다차원 공간에서 개체와 가장 근접하여 있는 이웃 개체를 비교하여 개체들을 분류한다. 이 알고리즘은 학습 데이터에 기반을 두어 새로운 개체가 분류기에 입력되면 몇 개의 근접한 개체를 비교하여야 하는지 측정하고 새로운 개체의 분류 정보를 정의한다. 본 논문에서는 145곡의 학습 데이터를 사용하였으며, 가장 가까운 이웃 개체 5개의 비교를 통해 새로운 개체를 정의하였다. 그리고 클래스가 정의되지 않은 개체가 새로이 입력되면 이 개체의 위치를 다차원 공간에 전사하고 그것과 가장 가까이에 이웃하는 5개의 클래스가 정의된 개체와 Euclidean 거리 계산을 통해 4개의 클래스 중 한 클래스로 개체를 정의한다. 본 논문에서 제안하는 시스템은 이 과정을 통해 음악의 무드를 분류 하여 사용자가 선택한 한 곡과 같은 클래스의 음악을 추천 시스템에 전달한다.

3.2 추천 시스템

무드 분류 시스템에서 전달한 곡들을 사용자의 재생 목록에 추가하기 위해 두 가지 모듈로 분리된 추천 시스템을 활용한다. 먼저 음악의 장르를 분류하기 위해서는 음악 신호 데이터로부터 적당한 모델을 생성해야 한다. 따라서 음악 추천 모듈에서는 음악 신호에서 MFCCs를 추출하고 HMM에 적용하여 모델을 생성한다. 이 방법은 음악의 장르를 분류하는 가장 대표적인 방법으로 Xi. Shao et al. [6]에 의해 제안 되었다. 본 논문에서는 이렇게 생성된 음악 모델을 기반으로 모델 간 마코프 모델을 분석함으로써 음악 간 유사도를 측정하고 그래프 클러스터링을 통해 유사한 음악 별로 분류하여 그 정확도를 확인하였다. 음악간 유사도는 다음 식 (1)을 통해 계산된다 [7].

$$S_{ij} = \frac{\frac{1}{N_i} [\log P(O_i|H_i) - \log P(O_j|H_i)]}{2} + \frac{\frac{1}{N_j} [\log P(O_j|H_j) - \log P(O_i|H_j)]}{2} \quad (1)$$

S_{ij} 은 음악의 유사도 값이고, 여기서 i 와 j 는 음악 i, j 를

의미한다. H_i 와 H_j 는 i, j 음악의 HMM을 의미한다. O_i 와 O_j 는 H_i 와 H_j 에서 가장 자주 되는 시퀀스이며, N_i 와 N_j 는 O_i 와 O_j 의 길이이다. (1)의 공식을 통해 각 곡들 간의 유사도 점수가 주어지고, MCL 그래프 클러스터링 알고리즘 [8]을 적용하여 분류의 정확도를 확인하였다. 음악 추천 모듈에서는 사용자의 재생 목록을 분석하여 사용자의 취향에 적합한 음악을 재생 목록에 추가한다. 이 과정은 사용자의 재생 목록에 저장된 음악의 그룹을 생성하는 과정을 포함한다. 음악들은 “얼마나 비슷한 장르”를 의미하는 유사도를 통해 그룹이 생성되고, 시스템이 음악을 추천할 때 사용자의 음악 장르적 취향을 재생 목록에 반영하기 위해 사용된다. 사용자의 재생 목록 분석 과정에서 음악들은 유사도를 기반으로 그룹을 나눈다. 그룹화를 위해 시스템은 사용자의 재생 목록의 음악 간 유사도를 표현하는 유사도 그래프를 생성한다. 그러면 시스템은 MCL 그래프 클러스터링 방법을 적용하여 음악을 유사한 음악들끼리 그룹화 한다. 음악 유사도 그래프는 재생 목록의 음악간 유사도 관계를 의미하고 사용자가 들었던 음악들의 유사도 그래프를 통한 클러스터링 결과와 같은 그룹을 생성한다. 따라서 사용자가 음악 장르적 취향 변화가 발생하여도 사용자의 재생 목록 분석을 통해 사용자의 변경된 취향 또한 음악 추천에 반영할 수 있게 된다. 본 논문에서는 효과적인 음악 그룹 생성을 위해 MCL 그래프 클러스터링 방법을 적용하였다. MCL 그래프 클러스터링 방법은 S. V. Dongen [9]에 의해 제안되었으며, 음악을 node로 음악간 유사도 정보를 edge로 하여 node 간 연결 정도와 edge의 수를 기반으로 클러스터링 과정을 수행한다. 음악 추천 모듈에서는 매번 추천 과정이 발생 될 때마다 클러스터링 과정을 수행 한다. 따라서 추천 초기에 사용자에게 적합한 음악이 추천 되지 못 한다 하더라도 이 과정을 통해 시스템은 사용자가 선호하는 음악을 시간이 지남에 따라 보다 정확히 추천하게 되어 그 추천 결과가 지속적으로 향상될 수 있다.

4. 실험 및 평가

본 연구에서 제안하는 시스템을 평가하기 위해 총 241 곡의 음악을 수집하였으며, 16,000 Hz, 16 비트, 모노 채널, wav 파일 형식으로 변환하였다. 또한, jAudio 툴을 사용하여 음악의 특징을 대표하는 12개의 요소를 추출하였다. jAudio 툴은 음악 신호로부터 특징을 추출하는데 일반적으로 사용되는 추출기로서 보다 자세한 정보는 jAudio 웹사이트 [10]에 기술되어 있다. jAudio 툴은 각 요소의 전체 평균과 표준편차를 추출하여 주므로 식 (2), (3)을 통해 통계적 계수와 그 정규화 값을 계산하였다.

$$Statistical\ Coefficient(SC_{kl}) = \frac{\sigma(F_{kl})}{ave(F_{kl})} \quad (2)$$

$$Nomalization(SC_{kl}) = \frac{SC_{kl}}{MAX(SC_l)} \quad (3)$$

SC_{kl} 는 통계적계수이며, F_{kl} 는 각 요소의 값이다. k 는 음악을 l 은 각 요소의 종류를 의미한다. 모든 요소 값은 0과 1사이의 값으로 정규화 되어 분류작업에 적용된다.

음악의 무드는 4개의 클래스 (A, B, C, D) 로 정의되며 k-NN에서의 분류 결과는 표 2에서와 같이 k=5에서 67.71%의 가장 높은 분류 정확도를 보였다. 따라서 추가되

는 새로운 음악은 Euclidean 거리 기반으로 가장 가까이 위치하는 5개의 음악들과 비교되어 그 클래스를 정의하게 된다. 추출한 음악 요소는 실험을 통해 음악의 무드를 가장 잘 정의하는 요소로 선별되어 시스템에 적용되었으며, 음악 재생 목록 생성 시스템에서는 식 (2), (3) 을 통해 계산된 값을 k-NN을 이용하여 음악의 무드를 분류하였다. 학습을 위해 145곡을 사용하였으며, 96곡이 검증 데이터로 사용되었다.

표 2. k-NN에서 k에 따른 오차율.

Table 2. Best k in k-NN.

k	학습 오차율	검증 오차율
1	0.00	39.95
2	19.60	36.05
3	27.39	34.42
4	25.88	35.05
5	22.37	32.29
6	25.88	36.68
7	26.63	35.05
8	31.40	36.05
9	25.88	32.79
10	28.14	34.42

표 3. k=5 에서의 오차율.

Table 3. Error Report in k-NN with k=5.

클래스	곡수	오차	오차율
A	26	4	15.38
B	21	7	33.33
C	22	9	40.91
D	27	11	40.74
전체	96	31	32.29

표 3에서와 같이 가장 좋은 분류 결과를 보인 클래스는 A이며, 클래스 C에서 가장 나쁜 결과를 보였다. 새롭게 추가된 음악은 시스템을 통해 정규화 되고 가장 가까이 있는 5개의 값과 비교하여 그 클래스가 정해지게 된다. 사용자는 자신의 상황에 적합한 음악을 한곡만 선택하고 그 음악과 동일한 클래스의 곡들이 추천 시스템으로 전송된다. 그리고 추천 시스템을 통해 사용자의 재생 목록에 추가 된다. 사용자는 자신의 취향에 맞는 장르의 음악만을 선택하게 되고 자신의 원하지 않는 음악은 재생 목록에서 제거한다. 이 과정을 통해 얻어진 사용자의 재생 목록과 추천 목록의 결과는 표 3과 같다.

표 3에서의 추천 결과와 같이 Electronic 장르를 선호하는 사용자 (87.5%)는 Electronic 음악을 다른 장르의 음악보다 더 많은 비율(69.6%)로 추천 받은 것을 확인할 수 있다. 특히 음악적 취향이 Jazz로 분석된 사용자 (84.5%)는 73.3% 비율로 Jazz 장르의 음악을 추천 받아 가장 높은 추천 정확도를 보였다. Metal 이나 Rock 장르를 선호하는 사용자는 추천 정확도가 비교적 낮다. 이것은 두 장르 간의 유사도가 상당히 비슷하여 클러스터링에 어려움을 보인 결과이다. 반면 Pop 장르는 장르적 특성이 불분명하여 가장

낮은 정확도를 보였다.

표 4. 사용자의 음악적 장르 취향에 따른 음악 추천 결과.
Table 4. Result of user play-list.

재생 목록	장르 (%)					
	Elec.	Metal	Jazz	Pop	Rock	Others
사용자 재생 목록	87.5	72.5	84.5	84.0	83.5	-
추천 재생 목록	69.6	52.8	73.3	55.1	63.2	-

다음 과정은 Rock 장르를 선호하는 사용자에게 시스템이 실제로 재생 목록을 생성해 주는 과정이다.

(과정 1) : 사용자는 클래스 A의 곡을 선택

- 사용자의 재생 목록
 - 곡명 : Bring The Noise, 클래스: A, 장르: Metal
- 추천 재생 목록
 - 곡명 : Bring The Pain, 클래스: A, 장르: Pop
 - 곡명 : Fade To Black, 클래스: A, 장르: Metal
 - 곡명 : My Generation, 클래스: A, 장르: Rock

시스템은 클래스 A인 3곡을 사용자에게 추천

(과정 2) : 사용자는 시스템이 추천한 3곡 중 자신의 음악 장르적 취향인 Rock 장르를 주로 선택

- 사용자의 재생 목록
 - 곡명 : Bring The Noise, 클래스: A, 장르: Metal
 - 곡명 : My Generation, 클래스: A, 장르: Rock
- 추천 재생 목록
 - 곡명 : Seasons in the abyss, 클래스: A, 장르: Rock
 - 곡명 : In Bloom, 클래스: A, 장르: Rock
 - 곡명 : Personality Crisis, 클래스: A, 장르: Metal

시스템은 반복적으로 클래스 A의 음악 중 사용자의 재생 목록에 있는 음악과 비슷한 장르의 음악을 추천

(과정 3) : 사용자는 자신이 듣고 싶은 장르의 음악을 주로 선택

- 사용자의 재생 목록
 - 곡명 : Bring The Noise, 클래스: A, 장르: Metal
 - 곡명 : My Generation, 클래스: A, 장르: Rock
 - 곡명 : Seasons in the abyss, 클래스: A, 장르: Rock
 - 곡명 : In Bloom, 클래스: A, 장르: Rock
- 추천 재생 목록
 - 곡명 : Welcome to the Jungle, 클래스: A, 장르: Rock
 - 곡명 : Fight the Power, 클래스: A, 장르: Rock
 - 곡명 : DU HAST, 클래스: A, 장르: Metal

시스템은 클래스 A인 음악 중 사용자의 재생 목록에 저

장된 곡들과 유사한 음악을 추천

(과정 3) : 사용자는 듣고 싶은 음악만을 남기고 제거

- 사용자의 재생 목록
 - 곡명 : Bring The Noise, 클래스: A, 장르: Metal
 - 곡명 : My Generation, 클래스: A, 장르: Rock
 - 곡명 : Seasons in the abyss, 클래스: A, 장르: Rock
 - 곡명 : In Bloom, 클래스: A, 장르: Rock
 - 곡명 : Welcome to the Jungle, 클래스: A, 장르: Rock
 - 곡명 : Fight the Power, 클래스: A, 장르: Rock
 - 곡명 : DU HAST, 클래스: A, 장르: Metal

- 추천 재생 목록
 - 곡명 : Paradise City, 클래스: A, 장르: Rock
 - 곡명 : Street Fighting Man, 클래스: A, 장르: Rock
 - 곡명 : Bring the Noise, 클래스: A, 장르: Rock

동일 클래스의 음악간 유사도 분석과 사용자의 재생 목록 분석을 통한 추천과정 계속

(과정 4) : 추천 완료된 재생 목록

- 사용자의 재생 목록
 - 곡명 : Bring The Noise, 클래스: A, 장르: Metal
 - 곡명 : My Generation, 클래스: A, 장르: Rock
 - 곡명 : Seasons in the abyss, 클래스: A, 장르: Rock
 - 곡명 : In Bloom, 클래스: A, 장르: Rock
 - 곡명 : Welcome to the Jungle, 클래스: A, 장르: Rock
 - 곡명 : Fight the Power, 클래스: A, 장르: Rock
 - 곡명 : DU HAST, 클래스: A, 장르: Metal
 - 곡명 : Paradise City, 클래스: A, 장르: Rock
 - 곡명 : Street Fighting Man, 클래스: A, 장르: Rock

위의 추천 과정을 살펴보면 먼저 사용자는 과정 1을 통해 자신이 놓인 상황에서 가장 듣고 싶은 음악을 한 곡 선택하였다. 그러면 시스템은 이와 동시에 선택된 곡과 같은 클래스 (A)의 음악들을 추천 시스템에 입력하게 된다. 그리고 이 곡과 가장 유사한 장르의 음악을 추천 재생 목록에 저장한다. 사용자는 추천된 음악 중 자신의 장르적 취향과 동일한 음악을 선택하고 나머지는 제거하여 재생 목록을 채워 간다. 위 과정의 반복적인 수행으로 결국 과정 3에서 추천 재생 목록의 결과와 같이 사용자가 원하는 장르(Rock)만을 추천받을 수 있게 된다. 또한 마지막 사용자의 재생 목록과 같이 대부분 Rock 장르로 된 재생 목록을 얻게 된다.

5. 결론 및 향후 연구

음악 재생 목록은 그 생성이 번거로울 뿐만 아니라 사용자의 상황 및 취향을 반영해 주어야 한다. 본 논문에서 제안하는 시스템은 사용자가 자신의 상황을 반영하는 한 곡만을 선택하여 재생 목록을 작성하는 방법으로 재생 목록 작성의 번거로움을 해결하였으며, 사용자의 재생 목록 분석을 통해 음악을 추천하여 사용자의 취향에 적합한 음악을 추천하는 것을 확인하였다.

이 시스템을 모바일 기기나 웹 서비스에 내장한다면 사용자가 언제 어디서든지 자신의 원하는 음악을 제공받을 수 있을 것이다. 또한, 사용자의 환경에 대한 보다 많은 정보를 얻을 수 있게 되어 곡 선택 없이도 재생 목록을 자동으로 제공 받을 수 있을 것으로 기대된다.

참 고 문 헌

[1] John C. Platt, C. Burges, S. Swenson, C. Weare and A. Zheng, "Learning a Gaussian Process Prior for Automatically Generating Music Playlist," *In Proc. NIPS*, vol. 14, pp. 1425-1423, 2002.

[2] F. Deli'ege, T. B. Pedersen, "Using Fuzzy Lists for Playlist Management," *In Proc. MMM*, pp. 198-209, 2007.

[3] R. E. Thayer, *The Biopsychology of Mood and Arousal*, Oxford University Press, 1989.

[4] L. R. Rabiner, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York: Springer, 2006.

[6] X. Shao C. Xu and M.S. Kankanhalli, "Unsupervised Classification of Music Genre Using Hidden Markov Model," *IEEE Trans. on Multimedia and Exop*, vol. 3, pp. 2023-2026, 2004.

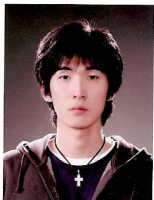
[7] K. Kim, D. Lee, T. Yoon and J. Lee, "A Music Recommendation System based on Preference Analysis," *In Proc. ICADIWT*, pp. 102-106, 2008.

[8] J. J. Aucouturier and Francois Pachet, "Music Similarity Measures: What's the Use?," *International Symposium on Music Information Retrieval*, pp. 157-163, 2002.

[9] S. V. Dongen, "Graph Clustering by Flow Simulation," *PhD thesis, University of Utrecht*, 2000.

[10] <http://jAudio.sourceforge.net>

저 자 소 개



방성우 (Sung-Woo Bang)
 2009년 : 성균관대 컴퓨터공학과 학사
 2009년~현재 : 동 대학원 전자전기 컴퓨터공학과 석사과정

관심분야 : 지능 시스템, 사용자 모델링, 음악 추천
 Phone : +82-31-290-7987
 Fax : +82-31-299-4637
 E-mail : sistone81@skku.edu



김태연 (Tae-Yeon Kim)
 2003년 : 공군사관학교 항공공학과 학사
 2009년~현재 : 성균관대 대학원 전자전기 컴퓨터공학과 석사과정

관심분야 : 웹 검색, 지능 시스템, 사용자 모델링
 Phone : +82-31-290-7987
 Fax : +82-31-299-4637
 E-mail : gtk2009@skku.edu



정혜욱 (Hye-Wuk Jung)
 1999년 : 한성대 정보공학과 학사
 2005년 : 성균관대 정보보호학과 석사
 2005년~현재 : 성균관대 전자전기 컴퓨터 공학과 박사과정

관심분야 : 패턴인식, 지능시스템, 정보보호
 Phone : +82-31-290-7987
 Fax : +82-31-299-4637
 E-mail : wukj@skku.edu



이지형 (Jee-Hyong Lee)
 1993년 : 한국과학기술원 전산학과 학사
 1995년 : 한국과학기술원 전산학과 석사
 1999년 : 한국과학기술원 전산학과 박사
 2002년~현재 : 성균관대 정보통신공학부 부교수

관심분야 : 지능시스템, 기계학습, 사용자 모델링
 Phone : +82-31-290-7154
 Fax : +82-31-299-4637
 E-mail : jhlee@ece.skku.ac.kr



김용세 (Yong Se Kim)
 1983년 : 서울대 기계공학과 학사
 1985년 : 스탠퍼드대 기계공학과 Design Division 석사
 1990년 : 스탠퍼드대 기계공학과 Design Division 박사
 1990년~1997년 : 일리노이대 조교수
 1997년~2000년 : 위스컨신대 부교수

2003년~현재 : 성균관대 기계공학부 교수
 2004년~현재 : 창의적설계기술연구소 소장

관심분야 : 서비스 통합시스템 융합디자인, Experience Design 지능형 지원 시스템 등 Design Cognition and Informatics 분야
 Phone : +82-31-299-6581
 Fax : +82-31-299-6582
 E-mail : yskim@skku.edu