Proceedings of KIIS Spring Conference 2008 Vol. 18, No. 1.

은닉 마코프 모델을 이용한 사용자 선호도 분석 기반의 음악 추천 시스템

Music Recommendation System Based on User Preference Analysis Using Hidden Markov Model

김건수[1], 이동훈[2], 윤태복[3], 이지형[4]

[1] 경기도 수원시 장안구 천천동 성균관대학교 전자전기컴퓨터공학과
E-mail: {kkundi[1], idoun[2], tbyoon[3]}@skku.edu
jhlee[4]@ece.skku.ac.kr

요 약

현재의 음악 서비스들의 대부분은 음악을 가수 이름이나 장르와 같은 키워드들로 구분하여 사용자에게 제공한다. 하지만 음악의 장르가 다양해지고, 장르별로 음악의 유형도 다양해짐에 따라 키워드 기반의 음악 제공 방법만으로는 사용자가 원하는 음악을 제공하는데 한계가 있다. 이런 한계점을 극복하기 위하여 음악 자체의 성질을 기반으로 음악을 분석하는 컨텐츠 기반의 음악 분석 방법이 필요하다. 또한 사용자가 원하는 음악을 제공 받을 수 있도록 사용자의 음악 선호도를 분석하여 그에 맞는 음악을 제공하는 방법도 필요하다.

본 논문에서는 음악의 시퀀스 정보와 특징을 추출하여 음악 모델을 구축하고, 이를 사용하여 사용자의 음악 선호도를 분석하는 방법을 제안하고, 사용자의 선호도에 맞는 음악을 제공하기 위하여 선호도 분석 방법을 통해 음악을 추천해주는 시스템을 제안한다.

Key Words : 음악 선호도, 은닉 마코프 모델, 음악 추천, MFCC

1. 서 론

다수의 사람들이 음악을 듣기 위해서 음악 제공 사이트들을 이용한다. 현재 웹상에는 여러 음악 제공 사이트[1][2]들이 존재하고, 이런 사이트들은 좀 더 다양한 방법으로 음악을 제공함으로써 사용자의 만족도를 높이기 위해 노력하고 있다. 하지만 현재 대부분의 사이트들은 장르 및 아티스트 정보와 같은 키워드를 기반으로 음악들을 분류하거나, 순위를 기반으로 음악들을 제공하고 있다. 사용자는 자신이 원하는 음악을 키워드를 통해서 검색하거나 사이트에서 추천해주는 음악을 선택하는 방법으로 음악을 듣는다. 음악을 추천하는 방법으로는 다른 사용자가 추천해 주는 음악을 제공하는 방법[1]과 사용자가 선호하는 장르를 선택하면 그 장르의 음악을 추천하는 방법[2]이 있다. 이런 방법들은 원하는 음악을 다른 사용자들이 추천한 음악들을 사용자가 직접 선별해야 하거나 단순히 같은 키워드로 구분되어 있는 음악을 추천해주는 한계점이 존재한다.

따라서 추천되는 음악의 정확성을 높이기 위해서는 음악 자체의 성질을 분석하여 사용자가 선호하는 음악을 분석하는 컨텐츠 기반의 음악 분석 방법이 필요하다.

컨텐츠 기반의 음악을 분석하는 기법은 소리의 파형으로부터 특질들을 추출하여 음악을 분석하는 기법이다[3]. 이 특질들은 사람들이 음악을 들을 때 인지하는 음의 높이, 음색, 리듬, 멜로디, 화성과 관련이 있다. 이 특질들 중 Mel Frequency Cepstral Coefficients (MFCC)는 음성 인식 분야와 음악의 분석에 사용되는 대표적인 특질이다[4][5]. G. Tzanetakis[6]와 X. Shao[7]은 음악의 장르를 분류하기 위하여 MFCC를 이용하였고, J.J. Aucouturier[8]는 음악의 유사도를 구하기 위하여 MFCC를 추출하여 사용하였다.

본 논문에서는 음악에서 추출된 MFCC를 사용해서 구축한 음악 모델과 사용자가 들은 음악 선호도를 기반으로 음악을 추천해주는 시스템을 제안한다. 음악 모델 구축에는 시퀀스를 고려하기 위하여 은닉 마코프 모델 기법을 사용하였고, 선호도 분석에는 사용자가 들은 유사한 노래를 그룹화하기 위하여 클러스터링 기법을 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는

전체 시스템의 구조와 추천을 위한 방법론은 설명하고, 3장에서는 2장에서 설명한 시스템에 대한 실험 방법 및 결과에 대하여 기술하고, 마지막으로 4장에서는 결론 및 향후 과제에 대하여 언급한다.

2. 선호도 기반의 음악 추천 시스템

선호도 기반의 음악 추천 시스템은 음악 모델링과정, 선호도 분석 과정, 음악 추천 과정으로 나눌 수 있다. 그림 1은 전체 시스템의 구조를 나타낸다. 음악 모델링 과정은 음악 데이터베이스에 있는 음악의 특징을 추출하여 모델링하고 저장하는 과정이다. 이 과정에서 구축된 음악 모델은 사용자 선호도 분석과 음악 추천에 사용된다.

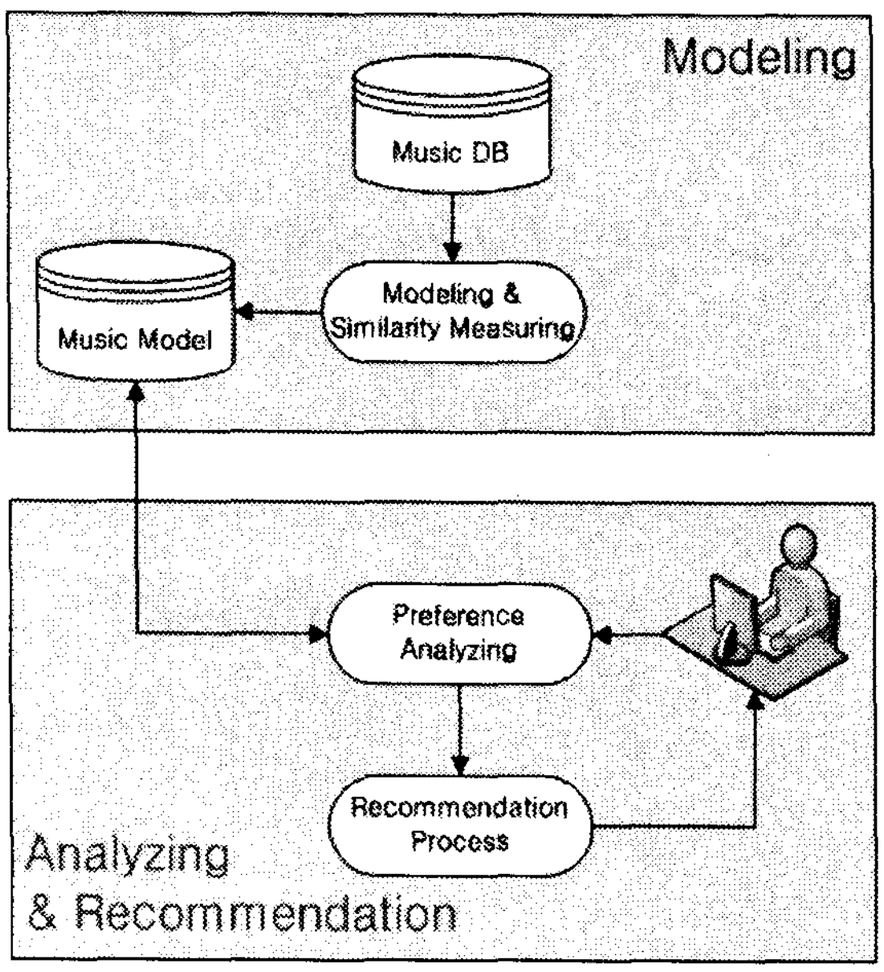

그림 1 음악 추천 시스템 구조

사용자 선호도 분석은 사용자가 들은 음악을 유사한 음악별로 구분하여 사용자의 선호도를 분석하는 과정이고, 음악 추천은 선호도 분석을 기반으로 선호도에 맞는 음악을 추천해주는 과정이다. 각 과정에 대한 자세한 내용은 아래에서 설명한다.

2.1 음악 모델링

시스템에서 사용자의 선호도를 분석하고, 선호도에 부합하는 음악을 추천하기 위하여 실제 음악 파일로 존재하는 데이터를 시스템에서 사용할 수 있게 모델링하는 과정이 필요하다. 이 과정에서 음악의 오디오 신호를 분석하여 음악별로 다른 음악들과의 유사도를 구한다. 우리는 하나의 음악과 다른 음악과의 유사도를 나타낸 유사도 벡터를 음악을 대표하는 음악 모델로 사용한다.

2.1.1 Mel Frequency Cepstral Coefficients

MFCC는 오디오 신호의 스펙트럼을 표시하기 위한 지수로써 주로 음성 인식을 위한 비선형 모델을 구축에 유용하게 사용된다[4]. 낮은 차수의 지수들은 스펙트럼의 변화에 따른 민감도가 낮은 반면 높은 차수의 지수들은 민감도가 높다. 따라서 우리는 음의 높이와 무관한 음악적 특질을 얻기 위해서 가장 낮은 8개의 차수만을 사용한다. 8개의 지수만 사용해도 음악의 특질을 효율적으로 나타낼 수 있다[8].

2.1.2. 음악 시퀀스(sequence)

음악은 여러 소리들의 연속적인 흐름이기 때문에 음악을 하나의 시퀀스로 볼 수 있다..

우리는 음악을 50ms 단위로 나누고, 각 단위 음악별로 MFCC를 추출하였다. MFCC의 8개의 차수만을 사용하므로 우리가 사용하는 음악의 시퀀스는 8차원 벡터들의 연속으로 이루어져있다.

2.1.3. 은닉 마코프 모델

은닉 마코프 모델은 상태 전이 확률, 각 상태에 대한 확률 분포와 초기 상태 확률을 가지고 있는 모델로 모델과 관측된 시퀀스간의 일치 정도를 확률 값으로 알려주거나 모델과 가장 유사한 시퀀스를 생성해주는 모델이다[9]. 새로운 은닉 마코프 모델은 시퀀스를 학습시킴으로써 얻을 수 있다.

우리는 학습에 사용하는 시퀀스로 음악 시퀀스를 사용한다. 하나의 음악 시퀀스는 하나의 은닉 마코프 모델을 생성한다. 이 은닉 마코프 모델을 통해 새로운 시퀀스를 생성하고 이 은닉 마코프 모델과 새로운 시퀀스로 음악의 유사도를 구한다.

2.1.4. 음악 유사도 및 음악 모델

음악의 유사도는 각 음악 시퀀스를 사용해 구축한 은닉 마코프 모델간의 거리를 구하는 식을 사용하여 구할 수 있다.

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

(1)식은 두 은닉 마코프 모델 간의 유사도를 구하는 식이다. H는 은닉 마코프 모델, O는

H에 의해 생성된 시퀀스, N은 O의 길이를 나타낸다. H_i, H_j가 유사할수록 S_{ij}의 값은 작아진다.

우리는 모든 음악 사이의 유사도를 구하고 이를 모델 구축에 사용한다. 하나의 음악은 (2)에서와 같이 다른 나머지 음악과의 유사도 벡터로 표현한다.

$$M_i = (S_{i1}, S_{i2}, \ldots, S_{i|M|}) \quad (2)$$

(2)에서 M이 하나의 음악을 대표하는 음악 모델이다. 유사도로 이루어진 이 음악 모델은 새로운 데이터베이스에 저장되어 사용자 선호도 분석과 음악 추천 과정에 사용된다.

2.2 사용자 선호도 분석

사용자 선호도 분석 과정은 사용자가 들은 음악을 기반으로 선호하는 음악 유형을 분석하는 과정이다. 이 과정을 통해 사용자가 들은 음악을 유사한 음악 유형별로 구분한다. 우리는 유형 구분을 위하여 클러스터링 기법을 사용하고, 선호도별로 음악 유형에 점수를 차등 부여 방법으로 사용자의 선호도를 분석한다.

2.2.1. 음악 유형 구분

사용자가 들은 음악을 유사한 음악으로 그룹 짓기 위하여 미리 구축한 음악 모델들과 클러스터링 기법을 사용한다. 사용자가 얼마나 다양한 유형의 음악을 들었는지 시스템은 모르기 때문에 사용자가 들은 음악 유형에 따라 동적으로 클러스터의 개수를 늘릴 수 있는 클러스터링 기법을 사용해야 한다. 우리는 클러스터의 요소간의 거리에 따라서 클러스터의 수를 조절할 수 있는 Improved K-means 기법[10]을 사용하여 음악을 그룹화한다.

2.2.2. 선호도 분석

우리는 사용자가가 들은 음악 유형 그룹 중 다수의 음악이 존재하는 그룹을 사용자가 더 선호하는 음악 유형이라고 정의하였다. 따라서 유형 그룹 별 음악의 수를 기반으로 가중치를 정하여 사용자가 선호하는 유형에 차등으로 점수를 부여한다.

$$L = \sum_{i=1}^{|C|} C_i \quad (3)$$

$$w_i = \frac{C_i}{L} \quad (4)$$

식(3)은 사용자가 들은 음악의 수를 나타낸다. 식(3)에서 C는 클러스터 안에 존재하는 음악의 수, L은 사용자가 들은 총 음악의 수이다. 식(4)는 클러스터별 가중치를 부여하는 방식이다. 이 가중치는 음악 추천을 위한 음악별 추천점수를 산정할 때 사용된다.

2.3 음악 추천 방법

시스템의 최종 목적은 사용자의 선호도를 기반으로 사용자가 원하는 음악을 추천하는 것이다. 시스템은 음악별로 추천 점수를 산정하고, 높은 추천 점수를 받은 음악들을 사용자에게 추천한다.

$$MSum = \frac{\sum_{m=1}^{C} (S_{m1}, S_{m2}, \ldots, S_{mN})}{C_i} \quad (5)$$

$$Ri = wi \bullet Inverse(MSum) \quad (6)$$

$$RList = MakeList(R_1, R_2, \ldots, R_{|C|}) \quad (7)$$

식(5)는 클러스터 안에 있는 음악 모델들의 벡터 합을 구하는 식으로 $MSum$은 클러스터에 존재하는 음악들의 정규화된 벡터 합을 의미한다. 식(6)의 $Inverse(MSum)$은 음악이 유사할수록 음악의 유사도는 낮은 값을 가지기 때문에 $MSum$의 모든 벡터 성분의 값을 역수로 변환한다. 따라서 R은 가중치를 적용한 클러스터와 모든 음악에 대한 유사도 벡터가 된다. 식(7)은 식(6)을 통해 구한 모든 클러스터별 음악 유사도 중 가장 높은 점수를 얻은 음악들을 선별한다. $MakeList(\ldots)$가 음악들을 선별하여 추천리스트로 작성하는 역할을 한다. 이 선별된 음악 리스가 최종적으로 사용자에게 추천된다.

3. 실험 및 평가

본 논문에서 제안한 시스템을 평가하기 위하여 4개의 장르(Jazz, R&B, Punk, Hip-hop)별로 각각 40곡씩 160곡의 음악을 가지고 실험을 진행하였다. 이 음악들을 모델 구축에 사용하기 위하여 모든 음악에 대해 1분 30초의 오디오 샘플을 만들어 MFCC를 추출하였다.

우리는 사용자가 한 장르만을 5곡 들었을 경우 시스템이 얼마나 정확하게 사용자가 선호하는 음악 유형을 파악하여 음악을 추천하는지에 관한 실험을 진행하였다. 5곡의 음악은 장르 안에서 무작위로 선별하였고, 추천되는 음악은 선별된 5곡은 제외한 나머지 음악들 중에서 추천되었다. 실험 결과는 상위 10곡의 음악이 속하는 장르를 백분율로 기술한 값으로 10번의 실험 결과를 평균화하여 기술하였다.

표 1은 사용자가 Jazz, R&B, Punk, Hip-hop

을 각각 5곡씩 들었을 경우 시스템이 추천하는 음악의 장르를 나타내는 결과이다. L은 사용자가 들은 음악, R은 추천한 음악의 장르를 의미한다.

L / R	Jazz	R&B	Punk	Hip-hop
Jazz	**80%**	13%	0%	2%
R&B	12%	**66%**	0%	5%
Punk	0%	0%	**99%**	0%
Hip-hop	8%	21%	1%	**93%**
합계	100%	100%	100%	100%

표 1 제안 시스템 결과

R&B에서는 조금 낮은 정확도를 보였지만, 대부분의 장르에서 사용자가 들은 유형의 음악을 추천했다.

우리는 실험 결과를 [10]에서 제안한 시스템과 비교해 보았다. 표 2는 우리가 제안한 시스템의 실험과 동일한 조건으로 실험한 결과이다.

L / R	Jazz	R&B	Punk	Hip-hop
Jazz	**90%**	0%	0%	17%
R&B	10%	**44%**	8%	55%
Punk	0%	11%	**82%**	0%
Hip-hop	0%	45%	10%	**28%**
합계	100%	100%	100%	100%

표 2 다른 시스템 결과

표 2와 표 1를 비교해보았을 때, 사용자가 선호하는 음악이 Jazz인 경우 우리가 제안하는 시스템에서의 정확도가 낮았지만, 전체적인 결과를 봤을 때, 정확도가 매우 향상되었음을 알 수 있었다.

4. 결론 및 향후 연구 과제

본 논문에서는 음악적 특질을 이용하여 사용자가 들었던 음악을 모델링하고, 이 음악 모델을 기반으로 사용자의 선호도를 분석하여 선호도에 맞는 음악을 추천하는 시스템을 제안하였다. 이 시스템은 음악이 가지고 있는 특질을 고려하여 실제로 사용자가 선호하는 음악 유형과 유사한 음악을 추천해줄 수 있다.

하지만 모든 음악에 대한 유사도를 구하여 시스템에 적용하기 때문에 음악의 수가 많아질수록 유사도를 구하여 음악을 모델링하는데 많은 시간이 소요되고, 음악 모델 자체의 크기가 커지는 단점을 가지고 있다. 향후 다수의 음악에 대해서도 시스템이 동작할 수 있도록 음악 모델 구축시 유사도를 구해야만 하는 음악들의 범위를 한정지을 수 있는 방법에 대한 연구가 필요하다.

감사의 글

본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 지식경제부의 유비쿼터스컴퓨팅및네트워크원천기반기술개발사업의 08B3-B1-10M 과제로 지원된 것임

참 고 문 헌

[1] http://www.bugs.co.kr
[2] http://www.muz.co.kr
[3] N. Scaringella, G. Zoia, and D. Mlynek, "Automatic Genre Classification of Music Content," IEEE Signal Proces Magazine, Vol. 23, no. 2, pp. 133-141, 2006
[4] L.R. Rabiner, "Fundamentals of speech recognition," Prentice-Hall, 1993
[5] B. Logan, "Mel Frequenct Cepstral Coefficients for music modeling," The First International Symposium on Music Information Retrieval, 2000
[6] G. Tzanetakis, "Musical Genre Classification of Audio Signals," IEEE Transactions on Speech and Audio Processing, Vol. 10, No. 5, pp. 293-302, 2002
[7] X. Shao, C. Xu, and M.S. Kankanhalli, "Unsupervised Classification of Music Genre Using Hidden Markov Model," IEEE International Conference on multimedia and expo, 2004
[8] J.J. Aucouturier and Francois Pachet, "Music Similarity Measures: What's the Use?," International Symposium on Music Information Retrieval, 2002
[9] L. Rabiner and B. Juang, "An Introduction to Hidden Markov Models," ASSP Magazine, Vol. 3, pp. 4-16, 1986
[10] D,M Kim, K.H Park, J.H Jung, and J.H Lee, "Personalized Music Recommendation System Using Improved K-means Clustering Algorithm," International Symposium on Advanced Intelligent Systems, pp. 945-948, 2007