

사용자 경험정보를 고려한 결정트리 기반 음악 추천 시스템

김유리^{0*}, 김성지*, 김정호**, 조재림**, 이동욱***, 김석진***, 전수빈*, 서동만^{00*}

*대구가톨릭대학교 컴퓨터소프트웨어학부

**대구가톨릭대학교 인공지능·빅데이터공학과

***대구가톨릭대학교 컴퓨터소프트웨어학과

kyuri0215@cu.ac.kr, seongjiger@cu.ac.kr, altair4357@cu.ac.kr, whwo2142@cu.ac.kr,
du137532@cu.ac.kr, kairos0@cu.ac.kr, marsberry@cu.ac.kr, sarum@cu.ac.kr

A Decision Tree-based Music Recommendation System Using the user experience

Yu-ri Kim*, Seong-gi Kim*, Jeong-Ho Kim**, Jae-rim Jo**,
Dong-wook Lee***, Seok-Jin Kim***, Soo-bin Jeon*, Dong-mahn Seo*

*School of Computer Software, Daegu Catholic University

**Department of Big Data Engineering, Daegu Catholic University

***Department of Computer Software, Daegu Catholic University

요 약

최근 IT 기술의 발달로 태블릿, 스마트폰과 같은 다양한 디바이스로 손쉽게 음악을 감상할 수 있다. 하지만 최근 이런 기술 발달과는 다르게 사용자가 원하는 음악을 검색하는 방법은 고전적인 형태에서 벗어나지 않고 있다. 기존 음악 검색 방법은 텍스트 기반, 내용 기반, 소비자 감성 기반의 음악 추천 검색 방법이 있으며 저장된 메타 데이터를 이용하여 사용자의 질의에 대한 결과만 제공할 뿐 사용자의 경험 정보를 고려하지 않는다. 그리고 기존 플랫폼들은 사용자가 최근 많이 들은 가수, 장르, 분위기를 종합하여 사용자에게 어울리는 음악을 추천을 할 뿐 사용자의 경험정보를 고려하여 음악을 추천하지는 않는다. 본 논문에서는 사용자의 경험 정보를 활용하여 사용자 맞춤형 음악 추천 시스템을 제안한다. 본 시스템은 사용자의 현재 기분 정보, 주변 날씨 정보 등을 입력 받는다. 이후, 경험 정보를 기반으로 결정 트리를 통해 사용자 요구 기반의 음악 추천 시스템을 구축하였다.

1. 서론

오늘날 IT 기술의 발달로 음악은 손쉽게 접할 수 있는 예술이 되었다. 다양한 디바이스로 음악을 감상할 수 있으며, 사용자 요청을 통해 듣고 싶은 음악을 빠르게 검색하여 감상할 수 있다. 대표적인 음악 정보 검색 방법은 메타 데이터 정보를 이용한 텍스트 기반, 내용 기반, 감성 기반 검색 방법이 있다 [1-4]. 메타 데이터 정보를 이용하여 음악을 검색하는 방법은 검색 키워드에 따라 결과를 제공하지만 사용자의 경험 정보를 고려하여 음악을 추천하여 제공하지는 않는다. 그리고 음악 제공 플랫폼들은 사용자가 최근 많이 들은 가수, 장르, 분위기를 종합하여 사용

자에게 어울리는 음악을 추천을 할 뿐 사용자의 경험 정보를 고려하여 음악을 추천하지는 않는다.

본 문제를 해결하기 위해 사용자의 경험정보를 기반으로 맞춤형 서비스를 제공하는 개인화 서비스가 증가하고 있다. 개인화 서비스 중 대표적인 예시로는 구글의 ‘네스트 허브 맥스’가 있다. 네스트 허브 맥스는 인공지능을 통해 사용자의 얼굴과 사용 상황 정보를 기반으로 사용자의 편의성을 위한 기능을 제공한다. 주요 기능으로는 사용자가 원하는 시간대에 원하는 행동에 맞추어 대중교통, 날씨, 음악 재생, 일정 등과 같은 기능이다. 이와 같이 개인화 서비스가 증가하고 있는 만큼 사용자의 경험, 환경, 요구에 맞춘 개인 서비스로 서비스 패러다임이 변하고 있다 [5-7].

본 논문은 사용자의 사용자의 감정, 주변 날씨 등

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학지원사업의 연구결과로 수행되었음"(2019-0-01056)

⁰⁰교신저자

을 활용한 음악 추천 시스템을 제안한다. 본 논문에서 제안하는 시스템은 결정 트리를 이용하여 사용자의 주변 경험 정보를 입력 받아 사용자에게 음악 리스트를 제공한다.

본 논문의 구성은 2장에서 기존 연구와 시스템 구현을 위한 관련 연구, 3장은 시스템에 대한 내용을 기술하고, 4장은 실험 환경, 5장은 실험에 대한 결과, 마지막으로 결론과 본 논문의 향후 연구 방향에 대하여 기술한다.

2. 관련 연구

기존 음악 검색 방법에는 텍스트 기반 검색 방법과 소비자 감성 기반의 음악 추천 방법이 있다. 이 방법의 경우 사용자의 경험을 고려할 수 없으며 사용자의 질의에 대한 결과만 제공한다. 또한 이 방법은 메타 정보같은 키워드를 알고 있더라도 유사한 멜로디를 가진 음악을 검색하는데 어려움이 발생한다 [1].

기존 음악 추천 시스템 중 사용자가 많이 사용하는 서비스는 대표적으로 애플 뮤직, FLO, 지니 뮤직이 있다. 애플 뮤직은 사용자가 좋아하는 장르, 아티스트를 바탕으로 분석하여 음악을 추천해준다. 또한 노래 장르, 국가 등 추가 카테고리를 선택하여 카테고리에 해당하는 음악을 추천한다 [8]. FLO는 사용자 취향인 아티스트, 장르를 바탕으로 음악을 추천한다. 해당 시스템은 사용자가 선택한 아티스트, 음원차트 정보, 장르 정보를 수집하여 해당 데이터를 기반으로 사용자에게 특화된 음악을 추천한다 [9]. 지니 뮤직은 장르, 상황, 감성, 시대, 날씨, 및 계절 등 다양한 태그 선택 기능을 제공하여 태그 조합에 따라 사용자 기호에 적합한 음악을 추천한다. 하지만 태그 선택 개수가 2개로 제한되어 있어 다양성이 부족하다 [10].

소비자 감성 기반의 음악 추천 연구 [11]는 음악 장르별 분위기를 선정하기 위해 감성에 대한 연구를 진행하였다. 그러나 연구 [11]은 음악에 대한 형용사 표본의 개수가 낮기 때문에 음악에 대한 키워드 형용사를 수작업으로 평가하여 연구를 진행하였다. 이와 같은 연구 방법은 한정된 시간에 많은 표본을 다루지 못하는 문제점이 존재하며, 적은 개수의 표본만으로 연구가 진행되었다.

표 1은 각 장르별 분위기 표이다. 본 논문에서는 연구 [11]의 표 1를 분위기 키워드 분류에 활용하였다.

본 시스템은 새로운 음악 추천 방법을 구현하기 위해 결정 트리 알고리즘을 이용했다. 결정 트리란 이산 값을 포함하는 유한 집합에서 레이블을 예측하며, 의사 결정 규칙과 그에 대한 결과들을 트리 구조로

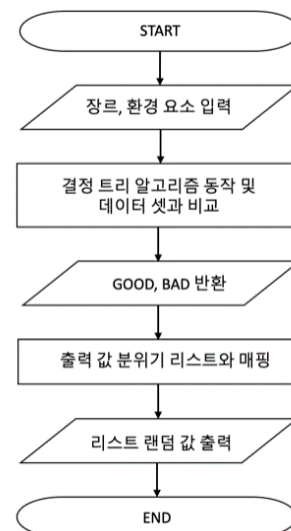
도식화한 의사 결정 지원 도구의 일종이다 [12]. 결정 트리는 질문을 던지면 그 질문에 대해 ‘예/아니오’ 형태의 답을 트리 구조를 이루고 있다. 질문에 대한 데이터를 분석하고, 이들 사이에 존재하는 패턴을 예측 가능한 규칙들의 조합으로 나타내는 것이다. 결정 트리는 ‘예’ 또는 ‘아니오’ 라는 결과 값을 반환한다.

3. 제안 시스템

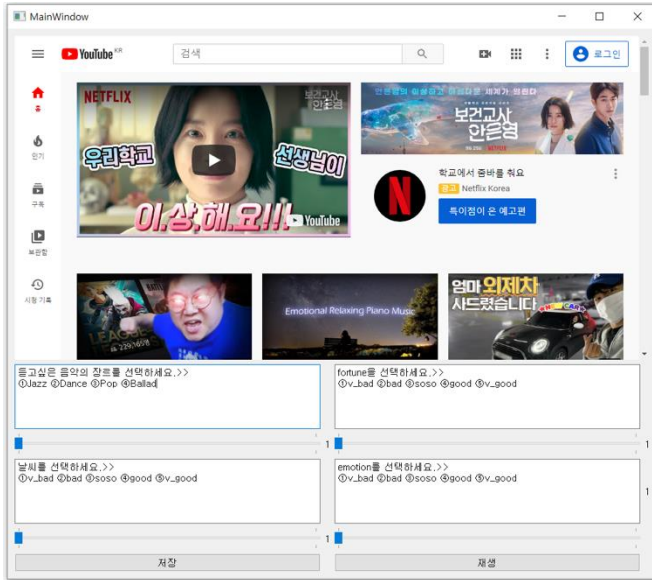
그림 1과 그림 2는 본 논문에서 제안하는 시스템의 흐름도와 사용자의 경험 정보를 점수 형태로 입력하는 UI를 보여준다. 시스템의 흐름도는 다음과 같다. 사용자가 슬라이드 바를 이용하여 사용자 경험 정보 점수를 입력한다. 입력한 데이터를 통해 시스템은 사용자 경험에 대하여 학습을 진행한 결정 트리 알고리즘 처리 결과를 반환한다. 반환 값과 표 1에 정의된 분위기 리스트와 매핑하여 유튜브에 검색할 키워드를 생성한다.

<표 1> 장르별 분위기 표

장르	결정 트리 결과값	
	GOOD	BAD
	분위기 종류	
Jazz	신나는, 경쾌한, 편안한, 평화로운	슬픈, 잔잔한, 우울한, 무거운, 차분한
Dance	신나는, 발랄한, 흥겨운, 경쾌한	슬픈, 잔잔한, 감각적인
Pop	신나는, 편안한, 흥겨운, 달달한	슬픈, 잔잔한, 쓸쓸한, 감성적인, 감성적인
Ballad	신나는, 달달한, 아름다운	슬픈, 잔잔한, 우울한, 감성적인



(그림 1) 시스템의 흐름도

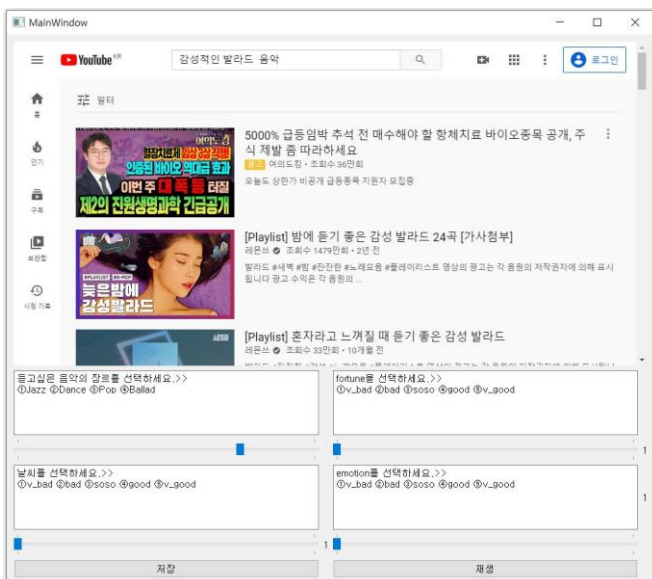


(그림 2) 음악 추천 시스템 UI

dataset

장르	날씨	하루 운세	감정	경험 정보의 평균 값	결과
jazz	v_bad	v_bad	v_bad	1	BAD
jazz	v_bad	v_bad	bad	1.33	BAD
jazz	v_bad	v_bad	soso	1.67	BAD
dance	bad	v_bad	bad	1.67	BAD
dance	bad	v_bad	soso	2	BAD
dance	bad	v_bad	good	2.33	BAD
pop	v_bad	good	v_good	3.33	GOOD
pop	v_bad	v_good	v_bad	2.33	BAD
pop	v_bad	v_good	bad	2.67	BAD
Ballade	v_bad	v_good	soso	3	GOOD
Ballade	v_bad	v_good	good	3.33	GOOD

(그림 3) 데이터 셋 샘플



(그림 4) 결정 트리 기반 음악 추천 시스템의 동작

그림 3은 결정 트리를 학습하기 위한 사용자 경험 데이터 셋의 일부분이다. 본 논문의 데이터 셋의 카테고리는 사용자의 경험정보에 영향을 미치는 기준을 고려하여 선별하였다. 사용자에게 추천되는 음악 리스트는 데이터 셋의 카테고리 중 날씨, 감정, 하루 운세의 평균 값을 통해 결정한다. 사용자가 원하는 장르의 음악 리스트를 추천하기 위해 음악 장르 카테고리를 추가하여 총 4개의 카테고리를 생성하여 데이터 셋을 구성하였다. 데이터 셋의 학습은 날씨, 감정, 하루 운세의 평균 값을 추출하기 위해 음악 장르 카테고리를 제외한 나머지 3가지 카테고리를 이용하여 결정 트리 알고리즘으로 학습을 시켰다. 서로 다른 날씨, 감정, 사용자 하루 운세의 값으로 이루어진 데이터 625개로 이루어져있고, 학습 시킨 데이터는 날씨, 사용자 하루 운세, 감정, 경험 정보의 평균 값으로 구성된다.

날씨, 감정, 하루 운세 카테고리별로 1~5 점 범위를 선정하고 1 점을 최하위, 5 점을 최상위 점수로 지정했다. 각 항목은 최하 점수를 v_bad(very bad), 최상 점수를 v_good(very good)으로 표기한다.

그림 3의 경험 정보의 평균 값은 날씨, 하루 운세, 감정 카테고리에 대한 평균 값이다. 예시로 그림 3 데이터 셋 샘플 일부분의 마지막을 참고하면 Ballade 장르의 경험 정보의 평균 값은 3.33이다. 이를 산출한 방법은 날씨 점수 v_bad(1 점), 하루 운세 점수(5 점), 감정 점수(4 점)의 평균이다. 평균 값에 따라 결과는 경험 정보의 평균 값에 따라 3점 이상이면 GOOD, 3점 미만이면 BAD 결과 값을 반환한다. 이와 같이 점수를 산정한 방식은 각 카테고리를 5 점을 최상위 점수로 지정하였기 때문에 평균 값인 3 점을 기준으로 3 점 이상이면 GOOD, 3 점 미만이면 BAD가 반환되게 적용하였다.

본 논문의 결정 트리는 예/아니오 대신 GOOD/BAD로 결과 값을 반환한다. 이와 같이 적용한 이유는 사용자의 주변 요소 점수를 입력 받아 긍정과 부정을 구분한다.

반환된 결과로 사용자가 선택한 장르와 표 1에 정의된 세부 분위기 키워드를 연결시켜 사용자 경험을 반영한 검색 키워드를 생성한다. 각 분위기 세부 키워드는 랜덤으로 추천이 된다. 세부 키워드는 연구 [3]의 사운드 디자인 작업의 기본 감성 공간과 핵심을 참고하였다. 각 장르별로 긍정과 부정의 분위기의 검색 결과가 다르기 때문에 직접 장르별로 검색 결과가 출력되는 분위기를 추려서 표 1로 정리를 하였다. 최종 단계로 생성된 추천 키워드를 통해 유튜브에서 사용자 경험을 반영한 음악을 추천한다.

4. 실험 환경 및 결과

제안 시스템의 구현을 위하여 i5-10210U CPU 와 16GB RAM 을 장착한 Window10 x64 운영체제를 사용하는 컴퓨터를 이용하였다. python3 버전을 이용하여 개발하였으며, UI 는 PyQt5 를 이용하여 구현했다. 사용한 라이브러리는 sklearn, pandas, PyQt5, PyQtWebEngine 를 사용하였으며, sklearn 는 결정 트리 알고리즘을 사용하기 위해, pandas 는 데이터 셋을 구성한 CSV 파일을 읽고 쓰기 위해서 사용하였다. 그리고 PyQt5, PyQtWebEngine 는 해당 시스템의 GUI 기능을 추가 하기 위해서 사용되었다.

시스템은 데이터 셋을 기반으로 사용자가 UI 를 통해 음악 장르를 선택하고 사용자 경험 정보의 점수를 매기고 저장한다. 그림 4 는 음악 추천 시스템을 사용한 예시이다. 음악 장르로 발라드, 하루 운세는 v_bad 인 최하 점수(1 점), 날씨는 v_bad(1 점), 감정을 최하 점수(1 점)를 매겼다. 사용자 경험 정보에 대한 평균 값은 3 점 미만으로 결정 트리 값이 BAD 가 나왔다. 시스템은 Ballad 장르 키워드에 결정 트리 결과 값 BAD 에 해당하는 분위기 종류에서 ‘감성적인’ 이란 키워드를 선택하여 그림 4 와 같은 결과를 출력한다.

출력 결과에서 보는 바와 같이 사용자가 입력한 장르와 경험정보 값을 통해 그림 4 와 같은 감성적인 발라드 음악이라는 결과가 정상적으로 출력됨을 확인하였다. 그림 4 는 사용자가 선택한 장르, 경험정보 값을 통한 분위기가 텍스트 형식으로 저장되어 유튜브를 통해 검색이 되며, 사용자는 검색 결과에 따른 결과를 통해 자신이 원하는 음악 리스트를 선택하여 감상할 수 있다.

5. 결론 및 향후 계획

본 논문은 사용자의 경험정보 점수를 입력 받은 뒤 결정 트리 알고리즘을 이용하여 사용자에게 음악을 추천하는 시스템을 구현했다. 본 시스템은 기존 음악 검색 방법의 단점을 해소하고 음악 검색 방법의 다양성을 향상하였다. 기존 음악 검색 방법인 텍스트 기반, 내용 기반, 감성 기반 검색 방법에서 음악을 결정하는데 시간이 오래 소모가 되었다면 본 논문의 검색 시스템은 사용자가 자신의 환경 요소를 입력하여 사용자가 편리하고 빠르게 음악 리스트를 추천 받을 수 있다. 하지만 해당 시스템의 음악 리스트는 사용자가 선택한 경험정보를 기반으로 랜덤 분위기가 선택되어 음악이 추천되기 때문에 사용자가 정확히 원하는 음악을 추천 하지 못할 수 있다.

향후 계획은 데이터 셋을 보완하여 현재 카테고리 뿐만 아니라 다양한 카테고리를 추가하여 결과 값의

정확도를 향상시킬 계획이며, 현재 시스템의 평균값을 상대적인 중간 값으로 나타냈기 때문에 해당 부분을 보완하여 추후 평균값을 구하는 방법에 대한 연구를 진행할 계획이다. 그리고 현재 PyQT 로 개발한 UI 를 보완하여 기능과 디자인을 보완하고 모바일, 웹사이트 등을 개발하여 사용자의 접근성이 용이할 수 있게 할 계획이다.

참고문헌

- [1] 윤보국, 홍성용, 감성 기반 음악 검색 및 추천 시스템 설계, 한국정보과학회 학술발표논문집, 2011, 38(1D), 153-155.
- [2] 이학선, 안형준, 윤세진, 정희훈, 이영호, 텍스트 기반 실시간 음악 편집 시스템, 한국정보과학회 학술발표논문집, 2015, 1889-1890.
- [3] 김태현, 이주환, 사운드 디자인 작업을 위한 감성기반 음원 검색 모델, 한국 HCI 학회 학술대회, 2017, 984-987.
- [4] 한국정보통신기술협회. 내용 기반 음악 검색을 위한 참조 모델, 2017
- [5] 전진배, 안세윤, 1인 가구를 위한 단독주택의 스마트홈서비스 적용 방안 연구, 한국콘텐츠학회논문지, 2020, 20(1), 180-191.
- [6] 황용석, 김기태개인화 서비스 진전에 따른 자동 추천 시스템 연구 동향과 방법론적 특성 연구. 사이버커뮤니케이션학보, 36(2), 221-253.
- [7] 최원석, 차서영, 최세정, 인지된 개인화와 종결 욕구가 소비자의 큐레이션 쇼핑 서비스이용 의도에 미치는 영향, 한국광고홍보학회, 2018, 89-122.
- [8] AppleMusic, Google Chrome 85 Brings Faster Graphics [Internet], <http://www.applemusic.co.kr/>
- [9] FLO, Google Chrome 85 Brings Faster Graphics [Internet], <https://www.music-flo.com/>
- [10] Genie, Google Chrome 85 Brings Faster Graphics [Internet], <https://www.genie.co.kr/>
- [11] 이승준, 서봉균, 박도형, 소비자 감성분석 기반의 음악 추천 알고리즘 개발, 한국지능정보시스템학회 학술대회논문집, 2018, 24(4), 106-106.
- [12] Freund, Yoav, and Llew Mason. "The alternating decision tree learning algorithm." icml. Vol. 99. 1999.