http://www.jsebs.org
ISSN: 2288-3908

The Journal of Society for e-Business Studies
Vol.22, No.3, August 2017, pp.75-85
https://doi.org/10.7838/jsebs.2017.22.3.075

청취 순서 성향을 고려한 랜덤워크 음악 추천 기법과 실험 사례

Experimental Study on Random Walk Music Recommendation Considering Users' Listening Preference Behaviors

최혜진(Hye-Jin Choe)*, 심준호(Junho Shim)**

초 록

전자 상거래 산업에서 많이 사용되고 있는 개인화 추천은 많은 분야에서 효과를 입증하고 있다. 개인화 추천을 위해서는 개인 정보를 포함하여 아이템을 재 분류해야하는 추가 작업이 필요하다. 본 연구에서는 개인 정보를 사용하지 않고 아이템을 재분류 하지 않는 추천 기법에 대해 제안한다. 음악 추천 영역으로 제한하여 실험하였으며, 실제 청취 이력 데이터를 사용하였다. 실험 분석을 통해 적은 데이터로도 유의미한 추천을 이끌어 낼 가능성을 살피고, 상황별 추천을 위한 아이템 수 분석과 추가 기법을 제안한다.

ABSTRACT

Personalization recommendations have already proven in many areas of the e-commerce industry. For personalization recommendations, additional work such as reclassifying items is generally necessary, which requires personal information. In this study, we propose a recommendation technique that neither exploit personal information nor reclassify items. We focus on music recommendation and performed experiments with actual music listening data. Experimental analysis shows that the proposed method may result in meaningful recommendations albeit it exploits less amount of data. We analyze the appropriate number of items and present future considerations for contextual recommendation.

키워드 : 음악 추천, 개인화 추천, 협업 필터링, 마르코프체인, 패턴 분석
Music Recommendation, Personal Recommendation, Collaborative Filtering, Markov Chain, Pattern Analysis

본 연구는 숙명여자대학교 교내연구비 지원에 의해 수행되었음(과제번호 1-1503-0120).
본 논문은 2017년도 한국전자거래학회&한국스마트미디어학회 춘계학술대회에서 발표된 "시간정보를 고려한 랜덤워크 음악 추천 기법" 논문이 우수 논문으로 추천되어 확장 수정된 것임.

* First Author, Department of Computer Science, Sookmyung Womens University (hyejin.choe@sookmyung.ac.kr)
** Corresponding Author, Department of Computer Science, Sookmyung Womens University (jshim@sookmyung.ac.kr)

Received: 2017-06-25, Review completed: 2017-08-02, Accepted: 2017-08-111

1. 서 론

빅 데이터를 처리 할 수 있는 기술이 발전되면서, 개인의 취향을 고려한 추천은 이미 전자상거래 산업에서 이윤을 내기위해 많이 사용되고 있다. 단순히 많이 구매한 상품이나 유행하는 아이템을 추천해 주는 것을 넘어 개인의 성향을 고려한 추천이 각광받고 있다. 대표적인 예가 아마존(Amazon)에서 추천을 사용하여 아이템의 구매력을 증진시키는 사례이다[9].

주로 개인화 추천은 영화와 음악, 뉴스 등에서 사용된다[11]. 아이템은 각각이 다른 특성을 가지고 있고, 주로 사용자가 제공한 정보를 고려하여 추천 목록을 만든다[5]. 예를 들어 영화는 사용자가 채점한 별점을 사용하고, 음악은 사용자의 장르에 대한 선호도를 제공받아 추천을 만들어 낸다[1].

본 연구에서는 추천 아이템의 영역을 음악 아이템으로 제한하였다. 사용자의 동의를 받아야하는 개인 정보는 실험에 사용하지 않았다. 따라서 데이터가 많이 없는 초기 사용자에게도 추천이 가능하고, 신규 음악을 추가하는 것도 구현이 쉬운 장점이 있다. 비가 오는 장마철엔 에픽하이의 '우산'과 같은 음악이 상위 차트에 진입하는 것에서 단기간의 사용자들의 성향을 반영한 추천을 고려하게 되었다. 그리고 한곡만 듣지 않고 연이어 유사한 곡을 청취한다는 것에서 마르코프 체인의 랜덤워크 기법을 응용하여 사용자의 음악 청취 패턴을 분석하였다.

구체적으로 본 연구가 추구하는 추천 방식은, 1) 청취 이력 이외에 데이터는 사용하지 않고, 2) 초기 사용자에게도 추천이 가능하며, 3) 사용자의 변화되는 취향을 고려하고, 4) 새 음악 추가가 쉬워야하고 아이템을 재분류 하지 않으며, 마지막으로 5) 반복 추천이 가능해야 한다는 점 등을 개발 환경으로 목표하였다.

즉, 사용자의 정보보호를 위해 청취 이력 이외의 데이터는 사용하지 않는 것을 고려하였고, 이는 초기 사용자에게도 추천이 용이하게 만드는 방법이다. 무엇보다 새로운 음악 추가가 쉽도록 아이템을 장르와 같은 특성으로 재분류 하지 않았다. 최근에 청취한 음악을 실험에 사용하는 것으로 사용자의 변화되는 취향을 고려할 수 있도록 하였고, 음악 아이템 특성상 반복 추천이 가능하도록 알고리즘을 제안하였다.

2. 관련 연구

본 연구는 추천을 위해 다른 사용자들의 데이터를 사용한다는 점에서 협업 필터링(Collaborative Filtering)으로 분류 할 수 있다. 협업 필터링은 크게 메모리 기반의 협업 필터링, 모델 기반의 협업 필터링, 하이브리드 추천으로 나뉜다[11].

메모리 기반의 협업 필터링은 추천이 필요한 타겟 사용자와 유사한 성향을 가진 사람을 찾아 그 사람이 좋아하는 아이템을 추천에 사용하는 방식이다. 추천하려는 아이템의 특성을 고려할 필요가 없기에 널리 사용된다[11]. 모델 기반의 협업 필터링은 사용자의 데이터베이스를 사용하여 모델을 추정 및 학습 한 다음 예측에 사용한다. 내용 기반 추천으로 분류되기도 하는 하이브리드 방식은 타겟 사용자가 즐겨찾는 아이템과 유사한 아이템을 찾아 추천해주는 방식이다. 유사한 아이템을 찾기 위해 아

이템의 특성을 추가적으로 분류하는 작업을 거쳐 추천에 사용해 준다.

청취 순서를 고려한 추천으로, Cooper et al.[3] 방법은 사용자가 채점한 별점(rating)으로 매트릭스를 구성한다. 마코프 체인 방식을 사용하며, 3번 전이를 하는 방식인 P^3 랜덤워크 방법을 사용해 높은 예측율을 자랑한다. 그러나 영화를 아이템으로 선정하여 계산한 Cooper et al.[5] 방법은 본 논문과 아이템의 차이를 보인다. 또한 3전이를 거치며 계산 시간이 오래 걸리는 단점을 보완하여 전의를 한 번만 진행하였다는 것에서 차이가 있다.

본 연구에서는 음악에 한정하여 추천 기법을 개발하였다. 흔히 음악 추천은 사용자로부터 음악 장르에 대한 성향을 제공 받아 실험에 사용한다[1]. 정보란 음악적 특징과 문화, 곡의 분위기, 리듬, 템포, 악기를 포함하여 그 사용자의 음악 선호도를 나타낸다고 본다. 이를 피어슨 상관관계를 거리로 측정하여 유사한 사용자를 찾는 방식이다. 이 방식은 개발자가 분류한 장르를 사용하고 사용자의 선호도를 미리 설문 조사해야한다는 단점이 있다. 또는 곡의 분위기를 태그로 분류하여 '슬픔', '신남', '아픔', '그리움'에 관련된 감정과 계절과 날씨를 반영하여 169개의 태그로 분류하여 곡 추천을 하는 방법도 있다[8]. 이런 방법들은 '음악'이라는 아이템에 집중 한 추천이다. 본 실험에서는 음악을 재분류 하지 않고, 유사한 사용자를 찾지도 않는다는 점에서 위 실험과의 차이를 가진다.

또 다른 음악추천으로 시간을 고려한 방법이 있다. 음악 특성과 사용자의 성향을 6개월 단위로 제공받아 성향의 정도가 비슷한 그룹으로 나누어 추천해 주는 방식이다[4]. 또는 온전히 기간만 고려하여, 계절과 월, 요일로 분류하여 시간 상황을 정보로 표현한 방법도 있다[7]. 이 방법은 같은 사용자라 할지라도 추천을 요청하는 시간에 따라 다른 추천을 할 수 있도록 요청한 시간을 고려하는 추천 방법이다. 이 역시 기간에 따라 분류했을 뿐, 유사한 사용자를 찾아 추천을 한다. 이 점에서 본 논문과의 차이를 가진다.

마지막으로 Choe[2] 논문에서는 기간과 청취 순서를 고려한 음악 추천 방식을 소개하는데, 본 연구와 관련된 다각적인 관련 연구 조사와 상세한 이론을 소개하고 있다. 반면 본 문은 구체적인 실험 기법과 사례에 한정하고, 청취 순서 성향을 분석한 실험 결과를 포함하고 있다는 차이점이 있다.

3. 제안 알고리즘

최근 성향을 고려하기 위해 실험 데이터를 최근에 청취한 순서대로 일정한 개수를 추출하였다. 여러 번의 실험을 하기 위해 월(Month)별로 나누어 실험을 실시하였다. 전체 파일에서 음악 아이디가 존재하지 않는 곡은 실험에 사용하지 않았다. 추천을 받을 타겟 사용자가 최근에 청취한 이력에서 실험 달에 해당하는 곡을 추출한다. 그런 후 타겟 사용자의 가장 최근 청취 기록 K곡을 사용하여 테스트 데이터로 만들었다. 데이터는 시간 역순으로 정렬되어 있기 때문에 사용자당 해당 개수를 채우면 건너뛰었다.

다음 <Figure 1>은 이를 의사코드로 나타낸 것이다. 추천을 받을 사용자는 'TargetUser'로 해당 아이디를 가진 사용자 데이터 'k'개를

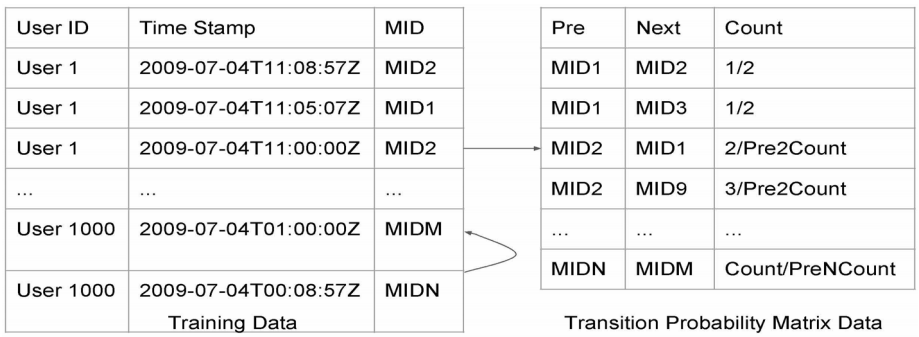
```
1: function DivideFile (TargetUser, TargetDate)
2: Input: Users History Text File (UserID, TimeStamp, MusicID)
3: Output: TrainingData and TestData txt file
4: K = Number of Target user's listening history for compare to result
5: Im = Number of Target user's listening history for training exclusive of K
6: n = Number of users' listening history for training exclusive of target user
7: while(OriginalFile != EOF){
8:   if MusicID == '': continue
9:     if UserID == TargetUser:
10:       if TimeStamp == TargetDate:
11:         if TestCount < K :
12:           TestCount += count
13:           write data in TestData
14:         else if TrainingCont < Im :
15:           Write data in TraningData
16:         end if
17:       end if
18:     else :
19:       if TimeStamp == TargetDate:
20:         if TraingCount < n:
21:           write data in TraningData for each users
22:         end if
23:       end if
24:     end if
25:  end if
26: end while
27: end function
```

〈Figure 1〉 Divide Test File and Training File

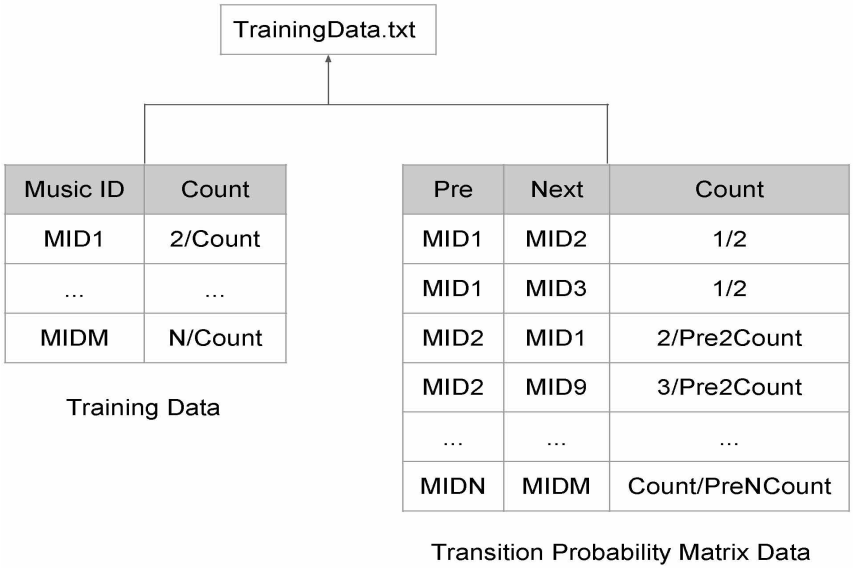
추출하여 테스트 데이터로 만들게 된다. 이때, 테스트 하고자하는 기간(달)을 고려하여 데이터를 추출한다. 테스트 데이터를 제외한 기록 중 추천이 필요한 타겟 사용자의 데이터 'Im'개와 타겟 사용자 이외의 사람들의 데이터 'n'개를 포함하여 트레이닝 데이터를 만든다. 트레이닝 데이터와 테스트 데이터는 사용자 아이디, 타임스탬프, 곡 아이디를 포함하여 텍스트 파일로 저장된다.

Im과 n곡을 더하여 청취 리스트 N곡이 만들어 진다. Im곡으로는 1행 N열 초기단계 매트릭스를 만들었고, 다른 사용자들의 데이터로는 N행 N열 추이 확률 매트릭스를 만들었다. 추이 확률 매트릭스를 구성할 때, 전에 청취한 곡과 이후에 청취한 곡의 순서를 포함하여 만든다. 이전에 청취한 곡(Pre)이 'A'곡이고 이후에 청취한 곡이 'B'곡이었으면, 'A'-'B'의 순서를 기억하여 카운트 하는 것이다. 순서를 기억하기 때문에 'B'-'A'는 다른 것으로 카운트 한다. 이전에 청취한 곡의 출현 횟수를 카운트 된 값의 분모로 사용하였다. <Figure 2>와 <Figure 3>을 참고하기 바란다.

초기단계 매트릭스는 트레이닝 데이터에서 타겟 사용자의 데이터만 추출하여 만들고, 나머지 데이터로 추이확률 매트릭스를 구성한다. 앞서 언급한 것처럼 곡의 순서가 중요하기 때문에, 순서가 일치하지 않는 것은 다른 것으로 카운트 하였다. <Figure 3>에서 Pre에 해당하는 곡의 출현 횟수를 카운트하여 순서 조합의 분모로 사용해주었다.



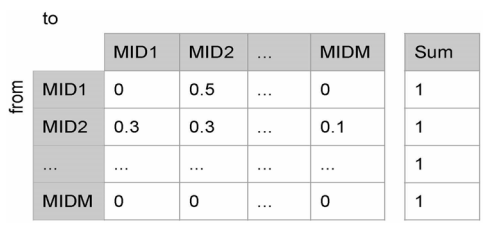
〈Figure 2〉 Transition Probability Matrix Value Calculated in Consideration of a Sequence



〈Figure 3〉 Initial State Matrix Data and Transition Probability Matrix Data

초기단계 매트릭스와 추이확률 매트릭스를 곱하여 1행 N열 결과 매트릭스를 만든다. 매트릭스의 값은 앞서 <Figure 3>처럼 계산된, 0~1사이의 변수 값으로 채워진다. 따라서 추이 확률 매트릭스는 <Figure 4>와 같은 모양으로 추이 확률 매트릭스가 구성되어야 한다. 행을 다 더한 값은 1이 된다.

<Figure 4>에서 추이확률 매트릭스의 행은



〈Figure 4〉 Transition Probability Matrix Configuration Value

이전에 청취한 곡이며, 열은 이후에 청취한 곡이다. 'MID1'을 이전에 청취하고 'MID2'를 이후에 청취했으면, 이전에 청취한 'MID1'의 출현 횟수를 카운트 하여 각각의 값에 나누어 준다. 이를 의사 코드로 나타내면 <Figure 5>와 같은 형태가 된다. 타겟 사용자가 청취한 Im곡과 다른 사용자들이 청취한 n곡을 합하여 N 크기의 리스트가 만들어진다. 따라서 MusicID는 N개 존재한다.

```
1: function MakeList(TrainingData)
2: Output: InitialList[], Transition[][]
3: MusicID = The total number of MusicID is N
4: PreMID = Previous music
5: NextMID = Next music
6: count = 0
7:   while(TrainingFile != EOF)
8:     if MusicID not in MidList[MusicID]
9:       MidList[MusicID] = 1
10:    else :
11:      MidList[MusicID] += 1
12:    count += 1
13:    end if
14:  end while
15:  for MusicID = 1 to N :
16:    InitialList[MusicID] = MidList[MusicID]/count
17:  end for
18:  while(TrainingFIle != EOF)
19:    if PreMID not in Transition[PreMID]:
20:      Transition[PreMID] = 1
21:      if NextMID not in Transition[PreMID][NextMID] :
22:        Transition[PreMID][NextMID] = 1
23:      else:
24:        Transition[PreMID][NextMID] += 1
25:      end if
26:    else:
27:      Transition[PreMID] += 1
28:    end if
29:  end while
30:  for MusicID = 1 to N :
31:    for MusicID = 1 to N :
32:      Transition[PreMID][NextMID] =
           Transition[PreMID][NextMID]/Transition[PreMID]
33:    end for
34:  end for
35: end function
```

〈Figure 5〉 Make MID List, Intial List and Transition List Pseudocode

그 후 두 매트릭스를 곱하여 준다. 결과 매트릭스로 나온 결과 값을 정렬하여 가장 높은 값을 가지는 곡을 추천에 사용한다. 평가 방법은 실험 데이터에서 가장 최근에 청취한 K곡을 추출하였고, 결과 데이터와의 일치성을 비교하였다.

4. 실 험

4.1 실험 데이터

실험 환경은 운영체제는 Windows 10 Home 이었으며, 64비트 이다. 8.00GB 메모리(RAM)을 사용하였고, 프로세서는 인텔(Intel(R) Core(TM) i5 -4460 CPU 3.20GHz)에서 진행하였다.

last.fm[10]의 실제 청취 데이터로 실험하였다. 데이터의 크기는 2.35GB였고, 사용자의 아이디(UID), 타임스탬프(Time Stamp), 아티스트 아이디(AID), 아티스트 이름(Artist Name), 음악 아이디(MID), 음악 이름(Music Name) 순으로 기록되어 있다. 이중 사용자 아이디와 타임스탬프, 곡 아이디를 실험에 사용하였고 곡 아이디가 없는 데이터는 제외하였다.

<Table 1>의 'Down' 곡처럼, 음악 아이디(Music ID)가 없는 곡들이 있다. 이런 곡들은 실험에서 제외하였다. 데이터는 한 사용자에 대한 기록이 끝난 후 다음 사용자의 기록이 이어진다. <Table 1>은 사용된 청취 이력 데이터 예시이다. User1의 2009년부터 2006년까지의 데이터가 이어진 다음 User 2에 대한 2009년 데이터가 이어져 있다.

〈Table 1〉 Example of 1,000 Users Listening History Data(Total Data)

User	Time Stamp	Artist ID	Artist Name	Music ID	Music Name
User1	2009-05-04T23:08:57Z	f1b1cf71-bd35-4e99-8624-24a6e15f133a	Troye Sivan	dc394163-2b78-4b56-94e4-658597a29ef8	Blue
User1	2009-05-04T21:03:51Z	6f3d4a7b-45b2-4c08-9306-8d271e92cb4f	Adele	fc89d5bf-a04c-41c9-836d-924e64302076	All I Ask
User1	…				
User1	2006-08-13T13:59:20Z	ce559a88-58ba-4d8a-8456-9177412d609c	OMI	cc7da9f6-df0e-4f88-a921-0f3b13f51fd1	Cheerleader
User2	2009-04-28T18:41:35Z	f1b1cf71-bd35-4e99-8624-24a6e15f133a	Troye Sivan	dc394163-2b78-4b56-94e4-658597a29ef8	Blue
User2	…				
User2	2006-02-24T18:05:42Z	eab8228c-a9c8-4ddb-a93d-ffca74a379bf	Marian Hill		Down
User3	2009-05-04T10:29:04Z	6f3d4a7b-45b2-4c08-9306-8d271e92cb4f	Adele	fc89d5bf-a04c-41c9-836d-924e64302076	All I Ask
User3	…				
User3	2005-10-30T22:23:21Z	ce559a88-58ba-4d8a-8456-9177412d609c	OMI	cc7da9f6-df0e-4f88-a921-0f3b13f51fd1	Cheerleader

사용자는 총 1,000명이었으며, 19,150,868줄이다. 기간은 2005년 02월 14일 00:00:07분을 시작으로 2013년 09월 29일 18:32:04분 데이터가 마지막이다. 각 사용자당 최소 1곡부터 최대 183,102곡을 재생한 사용자까지 다양했다. 107,528곡 중 아이디가 있는 69,420곡을 실험에 사용하였다. 평균적으로 한 사용자당 19,304곡을 재생했다. 만약 해당하는 달의 실험 데이터가 부족하면 그 전년도 데이터를 사용하였으며, 만약 그마저도 부족하면 참고할 수 있는 데이터만 추출하여 실험에 사용하였다.

4.2 실험 설정 값

실험은 파이썬 3.4에서 진행하였다. User1을 타겟 사용자로 설정하였고, 기간은 1월부터 12월까지 한 달씩 추출하여 실험하였다. 결과 예측율 비교하기 위한 테스트 데이터 K는 30곡으로 제한하였다. 초기 단계 매트릭스를 위한 User 1의 청취 이력은 각각 30곡, 50곡을 뽑았다. 추이 확률 매트릭스를 위한 데이터는 다른 사용자들의 청취이력으로 만들어 졌으며, 100곡과 150곡을 추출하여 실험해 보았다.

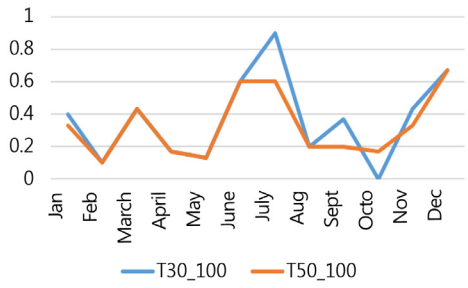
〈Table 2〉 Test Counts for Im and N

Name	Target user's training data number	Other users training data
T30_100	30	100
T50_100	50	100
T30_150	30	150
T30_300	30	300

T30_100과 T50_100은 매달 실험했다. 나머지 실험들은 최고 예측율과 최저 예측율을 보인 7월과 10월에 대해서만 실험을 진행하였다.

4.3 실험 결과

실험을 위해 달마다 타겟 사용자와 다른 사용자들의 데이터를 함께 추출하는데 걸린 시간은 평균 157초였다. 예측율은 추출해 놓은 K곡과 결과로 나온 음악 아이디의 일치성을 비교하였다. 다시 말해, 일치하는 곡의 개수를 K로 나누었고 이를 '예측율'이라 표현하였다.



〈Figure 6〉 Prediction Graph

결과 그래프를 살펴보면, T30_100은 90%의 높은 예측율을 보이는 달이 있는가하면, 0%로 낮은 예측율을 보이는 달도 존재한다. 반면 T50_100은 평균적인 예측율을 보이며, 최소 10%에서 최대 67%의 예측율을 보인다. 따라서 타겟 사용자의 최근 청취 이력이 50곡은 쌓여야 최소한의 예측율을 보장해 준다.

0%의 예측율을 보이는 달도 예측이 일치하지 않는 것이지 추천이 없는 것은 아니다. 실험으로 나온 곡의 개수를 살펴보면 <Table 3>과 같다. 일치하지 않는 곡과 사용자의 평가는 추후 이루어져야 할 과제이다.

결과를 살펴보면, 다른 사용자들이 999명 존재하기에 이들의 이력을 한 곡씩만 더 사용해도 계산 시간은 급격히 늘어난다. <Table 3>을 참고하면, 타겟 사용자의 데이터 30곡을 사

용하고 다른 사용자들의 데이터 150곡과 300곡을 각각 참고하여 매트릭스를 구성한 결과를 보면 예측율은 같은 반면 300곡씩 사용한 실험이 2배가 넘는 시간이 걸린 것을 확인할 수 있다. 이 결과로 타겟 사용자의 데이터가 많이 없는 상태에서 다른 사용자들의 데이터를 많이 사용한다고 예측율이 높아지는 것은 아니었다.

〈Table 3〉 Number of Results for Each Experiment, Average Time and best/Worst Prediction

Name	Average number of test results	Time (Min)	Best Pre-diction	Worst Predic tion
T30_100	50.8	85	0.90	0
T50_100	29.8	85	0.73	0.57
T30_150	11	156	0.37	0.03
T30_300	12	459	0.37	0.03

따라서 초기 사용자라면 다른 사용자들의 데이터를 최소 150곡씩은 사용하여 추천 매트릭스를 만드는 것이 좋고, 일반 사용자라면 타겟 사용자의 데이터는 50곡 이상씩, 다른 사용자들의 데이터는 100곡씩만 추출하여 추천 리스트를 만들어도 최소한은 보장하는 추천을 보인다.

5. 결 론

본 실험에서는 사용자의 개인정보 데이터를 사용하지 않고 오직 청취 이력 데이터만 사용하여 실험해 보았다. 따라서 처음 서비스를 사용하는 초기 사용자에게도 추천이 가능하다. 반면, 사용자의 개인정보와 장르 선호도 등을 사용하지 않았기에 다른 추천 알고리즘보다 상대적으로 낮은 예측율을 보인다. 하지만 아이템을 분류하지 않았기에 신규 곡에 대한 업데이트 없이 시스템을 운영할 수 있고, 사용자의 최근 성향을 반영하여 매트릭스를 구성하기 때문에 사용자의 변화되는 성향을 고려해 줄 수 있다. 따라서 실험 데이터가 많이 없는 신규 가입자나 처음 시스템을 개시한 곳에서 사용하기 좋을 것으로 기대한다.

낮은 예측율을 보완하기 위해서는 P^3 랜덤워크 기법[3]처럼 3번 전이를 실험 해 볼 필요가 있다. 이는 많은 계산을 필요로 하기에 싱글 머신에서 계산했을 경우 오랜 시간이 필요하다. 따라서 분산처리 환경에서 추가 연구가 필요하다. 또 다른 방법으로, 사용자에게 최소한의 정보만을 제공받는다는 연구 목표를 유지하기 위해 해당 곡을 '건너뛰기(Skip)'하는 정보를 활용하여 긍정/부정 성향을 만들어 낼 수 있다[10, 12]. 성향을 반영하여 Myung et al.[11]처럼 매트릭스를 구성하는 방법과, Yeon et al.[12]처럼 한번 이상 나타나는 곡은 긍정으로 간주하는 기법을 적용해 볼 수도 있다. 또한 곡의 청취 시간이 기준 쓰레숄드 이하로 매우 짧다면 해당 곡을 부정으로 간주하여 긍정/부정 성향을 청취 이력에서 추출하는 방법 등의 적용도 추후 연구로서 적용 가치가 있을 것이다.

References

[1] Bogdanov, D., Haro, M., Fuhrmann, F., Gómez, E. and Herrera, P., "Content-

based music recommendation based on user preference examples," Conference of The 4th ACM Conference on Recommender Systems, 2010.

[2] Choe, H. J., "Random Walk Music Recommendation Considering Users' Listening Preference Behaviors," Master's Thesis SookMyung Women's University, 2017.

[3] Cooper, C., Lee, S. H., Radzik, T. and Siantos, Y., "Random walks in recommender systems: Exact computation and simulations," Proceedings of the 23rd International Conference on World Wide Web, 2014.

[4] Farrahi, K., Schedl, M., Vall, A., Hauger, D., and Tkalcic, M., "Impact of Listening Behavior on Music Recommendation," The International Society of Music Information Retrieval, 2014.

[5] Hu, Y., Koren, Y. and Volinsky, C., "Collaborative filtering for implicit feedback datasets," Proceeding of the Eighth IEEE International Conference on Data Mining, 2008.

[6] Last.fm, http://www.last.fm/.

[7] Lee, D., Lee, S. K. and Lee, S. G., "Considering temporal context in music recommendation based on collaborative filtering," Proceedings of Korea Computer Congress, Vol. 36, No. 1, pp. 89-97, 2009.

[8] Li, J., Lin, H. and Zhou, L., "Emotion tag based music retrieval algorithm," Conference of Asia Information Retrieval Symposium, 2010.

[9] Linden, G. Smith, B., and York, J., "Amazon.com recommendations: Item-to-item collaborative filtering," IEEE Internet computing, Vol. 7, No. 1, pp. 76-80, 2003.

[10] Myung, J. S., Shim, J. H. and Suh, B. M., "Fast Random Walk with Restart over a Signed Graph," The Journal of Society for e-Business Studies, Vol. 20, No. 2, pp. 155-166, 2015.

[11] Su, X. and Khoshgoftaar, T. M., "A survey of collaborative filtering techniques," Journal of Advances in artificial intelligence, 2009.

[12] Yeon, J. H., Lee, D. J., Shim, J. H. and Lee, S. G., "Product review data and sentiment analytical processing modeling," The Journal of Society for e-Business Studies, Vol. 16, No. 4, pp. 125-137, 2011.

저 자 소 개

최혜진 (E-mail: hyejin.choe@sookmyung.ac.kr)
2015년 숙명여자대학교 컴퓨터과학 졸업 (학사)
2017년 숙명여자대학교 컴퓨터과학 졸업 (석사)
2017년~현재 KT 융합기술원 연구원
관심분야 개인화추천, 빅데이터, 패턴분석, 자연어처리

심준호 (E-mail: jshim@sookmyung.ac.kr)
1990년 서울대학교 계산통계학과 졸업 (학사)
1994년 서울대학교 계산통계학과 전산과학전공 (석사)
1998년 Northwestern University, Electrical and Computer Enginerring (박사)
2001년~현재 숙명여자대학교 소프트웨어학부 교수
관심분야 데이타베이스, 빅데이터, e-비즈니스 기반기술, 상품정보