소프트웨어 응용 최종 보고서

- SGD를 이용한 Recommendation System -

컴퓨터과학부 2015920062 홍성현

컴퓨터과학부 2016920027 오준영

`

1. 개요

우리 조는 사용자들이 기존에 매겼던 평점들을 이용하여 각 사용자들의 성향에 맞는 음악을 추천해주는 추천 시스템을 만들었다. 평점 예측을 위한 알고리즘으로는 모델 기반의 matrix factorization 기법을 사용하였고, 예측 평점 행렬을 학습시키기 위한 기법으로 SGD와 minibatch를 사용한 SGD를 구현했다. 또한 추천 목록에 나와있는 아이템들의 주요 키워드를 알려주기 위해 TF-IDF 기법을, 추천목록에 대한 사용자의 반응에 따라 예측 평점을 조절하기 위해 item-item CF 기법을 사용했다.

2. 지원 기능

우리의 음악 추천 시스템은 관리자 모드와 사용자 모드로 나눌 수 있으며, 각 모드는 다음과 같은 기능을 제공한다.

관리자 모드

- 기존에 주어진 평점 정보들을 통해서 예측 평점 행렬을 구할 수 있다.

  - SGD

  - 크기가 128인 minibatch를 이용한 SGD
- 미리 만들어 파일로 저장해 놓은 예측 평점 행렬을 불러올 수 있다.
- 현재 시스템이 저장하고 있는 예측 평점 행렬과 RMSE 값을 확인할 수 있다.
- 예측 평점 행렬을 구하는 과정에서 기록된 RMSE값을 확인할 수 있다.
- 우리가 만든 추천 시스템의 성능을 비교하기 위해, 단순한 Item-Item CF기법과의 RMSE 차이를 확인할 수 있다.

사용자 모드

사용자 모드는 시스템상에 예측 평점 행렬이 저장되어 있는 상태에서만 진입할 수 있으며, 진입할 때 사용자의 아이디를 입력한다.

- 각 사용자에 맞는 맞춤형 추천 목록을 확인할 수 있다. 이때, 각 음악별로 관련 키워드를 확인할 수 있다. 이 키워드 목록은 처음 프로그램을 실행할 때 계산되어 파일로 저장되며, 기존에 파일이 존재할 경우 그 파일을 로드하여 사용한다.
- 추천 목록에서 마음에 들지 않는 항목을 삭제할 수 있다. 이때, 삭제한 음악과 유사한 다른 음악들의 예측 평점도 내려간다. 변경 사항은 파일로 저장되어 프로그램을 종료해도 그 데이터가 유지된다.
- 인기 차트에서 흥미가 있는 음악을 선택하면 그와 유사한 다른 음악들을 추천받을 수 있다.

`

3. 추천 알고리즘 수행 방식 및 성능 비교

우리가 만든 추천 시스템에는 두 가지 추천 알고리즘 SGD와 minibatch를 사용한 SGD가 존재한다. 아래에 SGD의 수행과정을 간략히 나타내었다.

1. 주어진 평점 행렬에서 랜덤하게 1000개의 원소를 골라 test set으로 사용하고, 나머지를 train set으로 사용한다.
2. SVD를 통해 기존 평점 행렬을 2개의 행렬 Q, P^T로 분리한다. Factor 개수는 100개로 설정한다.
3. Train set의 원소들을 무작위 순서로 뽑아 Q, P^T, 상품 바이어스(b_x), 사용자 바이어스(b_i)를 학습시킨다. Object function으로는 아래의 식을 사용한다. 모든 Train set에 대해 Q, P^T를 학습시키는 것을 한 번의 epoch로 한다.

$$\sum_{x,i)\in R}\left(r_{xi}-(\mu+b_x+b_i+q_i\,p_x)\right)^2$$

goodness of fit

$$+\left(\lambda_1\sum_i\|q_i\|^2+\lambda_2\sum_x\|p_x\|^2+\lambda_3\sum_x\|b_x\|^2+\lambda_4\sum_i\|b_i\|^2\right)$$

regularization

그림 1. 강의 슬라이드 참조

4. 하나의 원소에 대해 Q, P^T를 학습시키는 것을 학습 횟수 1번이라 하자. 학습 횟수가 총 평점 수 / 4의 배수가 될 때마다 RMSE값을 구해준다. 이를 통해 한 번의 epoch당 4번의 RMSE를 구하게 된다. 미리 정해 놓은 학습 횟수를 달성하거나 이전에 구한 RMSE와의 차이가 10^{-5} 이하일 경우 학습을 종료한다.

Minibatch를 이용한 SGD는 각 원소마다 구한 기울기 값을 Q, P^T, bias에 바로바로 반영하지 않고 이를 모아둔 뒤 minibatch 크기만큼의 학습이 끝난 뒤에 한번에 반영한다는 차이점이 있다. 우리는 minibatch의 크기를 128로 설정했다. Minibatch를 사용한 SGD도 한 번의 epoch당 4번의 RMSE를 구하도록 조절해주었다.

`

예측 평점 행렬 학습에 일반적인 SGD와 batch 크기를 128로 한 SGD를 각각 5회씩 사용하여 구한 RMSE값과 이때 사용한 train set과 test set을 이용하여 구한 Item-Item CF의 RMSE 값을 비교하여 표로 정리했다.

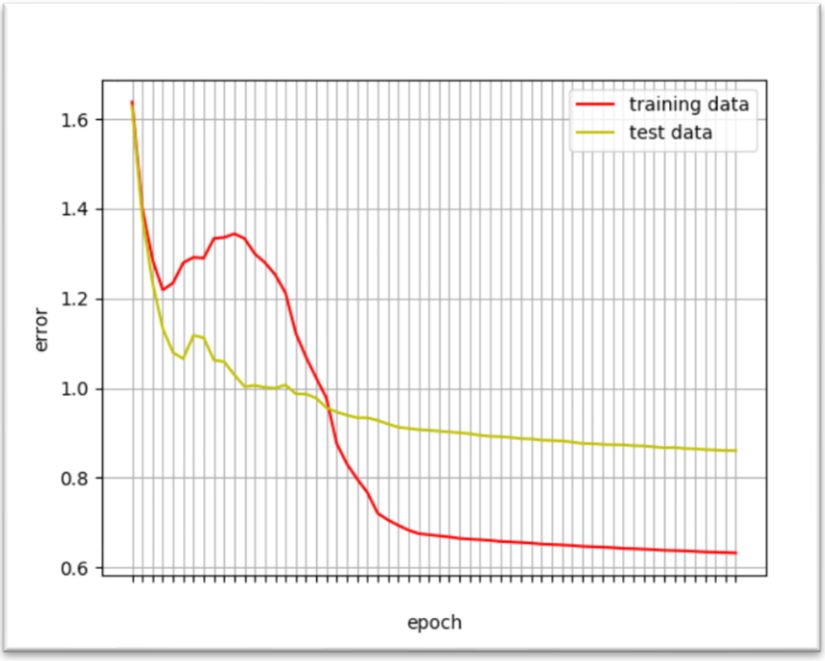
RMSE	1회차	2회차	3회차	4회차	5회차	평균
SGD	0.869	0.863	0.854	0.890	0.857	0.853
Minibatch	0.872	0.865	0.843	0.828	0.865	0.847
Item-Item CF	0.970	1.026	0.966	1.017	0.972	1.026

표 1. SGD 와 batch 크기가 128인 SGD, 일반적인 Item-Item CF의 RMSE 비교

수행시간(s)	1회차	2회차	3회차	4회차	5회차	평균
SGD	121.14	114.29	139.98	141.68	156.27	134.67
Minibatch	173.13	160.26	144.34	160.19	147.62	157.10

표 2. SGD 와 batch 크기가 128인 SGD의 수행 시간 비교

에러 그래프는 다음과 같은 형태로 나타난다.



`

SGD 기법을 통해 예측 평점 행렬을 구했으므로 예상대로 RMSE값이 진동하면서 특정 값으로 수렴하는 것을 확인할 수 있었다.

SGD와 minibatch를 이용한 SGD 둘 다 기본적인 Item-Item Collaborative Filtering보다 좋은 정확도를 가지는 것을 확인할 수 있었다. 두 방법을 비교해보니 SGD는 수행시간에 강점이 있고, Minibatch 기법은 평균 정확도가 더 좋은 것으로 나타났다.

4. 동작 시나리오

우리가 만든 추천 시스템은 사용자에게 세 가지의 시나리오를 가질 수 있다. 첫째로 특정 사용자가 자신을 위한 추천 목록을 보고 싶어할 때. 추천하는 음악의 ID와 함께 해당 음악과 관련된 키워드를 보여준다. 그 과정은 아래와 같다.

1. 자신의 아이디를 입력하여 사용자 메뉴로 들어가 첫 번째 메뉴 '추천 목록 출력' 을 선택한다.
2. 추천 받을 음악의 수를 입력하면 사용자가 아직 평점을 매기지 않은 음악들을 예측 평점이 큰 순서대로 정렬해서 보여준다. 각 항목 옆에는 예측 평점과 함께 관련된 키워드가 5개까지 표시된다.

```
1. user mode
2. admin mode
3. quit
select : 1
Enter your ID : AXFI7TAWD6H6X

 ----- user mode -----
1. 추천 목록 출력
2. 추천 목록에서 특정 항목 제거하기
3. 인기곡으로부터 추천받기
4. 돌아가기
select : 1
표시할 항목 수를 입력해주세요. : 10
---------------------------------------- 추천 목록 ----------------------------------------
---------- item ---- expected rate ------------------------- keyword ----------------------
1 .  |  B000003AEK  |   4.590    |     ['2pac', 'pac', 'rap', 'mama', 'die']
2 .  |  B004QSQM72  |   4.573    |     ['antony', 'vocals', 'talk', 'british', 'albatross']
3 .  |  B000004B9T  |   4.569    |     ['beautiful', 'modern', 'end', 'lisa', 'different']
4 .  |  B00000J2PJ  |   4.553    |     ['small', 'brings', 'word', 'years', 'sendak']
5 .  |  B000VI6T9C  |   4.517    |     ['heard', 'new', 'retro', 'nice', 'miss']
6 .  |  B00004U91U  |   4.501    |     ['girls', 'possible', 'art', 'ani', 'theyre']
7 .  |  B000002W5F  |   4.499    |     ['schmilsson', 'rock', 'recording', 'harrys', 'keith']
8 .  |  B00000189I  |   4.497    |     ['barmy', 'smith', 'punk', 'early', 'smiths']
9 .  |  B00006L3I6  |   4.493    |     ['soul', 'wallpaper', 'interesting', 'xavier', 'create']
10.  |  B00000053G  |   4.487    |     ['fast', 'know', 'clicc', 'camp', 'boot']
```

그림 3. 추천 목록 출력

`

관련 키워드는 각 음악에 등록된 사용자 리뷰들을 3전부 모은 뒤, TF-IDF 기법을 사용하여 결과값이 높은 단어부터 차례대로 나타냈다.

두 번째로, 사용자가 받은 추천 목록들 중 마음에 들지 않는 음악이 존재할 때, 그 음악을 추천 목록에서 제거할 수 있다. 이때, 삭제한 음악과 관련된 다른 음악들의 예측 평점을 유사도에 비례하여 내려준다. 동작 과정은 아래와 같다.

1. 자신의 아이디를 입력하여 사용자 메뉴로 들어가 두 번째 메뉴 '추천 목록에서 특정 항목 제거하기' 를 선택한다.
2. 기존에 '추천 목록 출력'을 수행한 적이 있었다면 그 목록을 화면에 표시하고, 아니면 원하는 수를 입력 받고 그 수만큼 추천 목록을 표시한다.
3. 추천 목록에서 제거하고 싶은 음악의 번호를 입력하면 그 음악은 추천목록에서 제거된다. 기존 평점 행렬에서 이 음악의 평점 값을 1로 만든다. 또한 이 음악과 20퍼센트 이상 유사한 음악은 유사도에 비례해서 예측 평점의 값을 줄여준다.

```
 ----- user mode -----
1. 추천 목록 출력
2. 추천 목록에서 특정 항목 제거하기
3. 인기곡으로부터 추천받기
4. 돌아가기
select : 2
Enter number of items : 20
---------------------------------------- 추천 목록 ----------------------------------------
---------- item ---- expected rate ------------------------ keyword ---------------------
1 .  |  B00000050R  |   4.469    |     ['banks', 'hort', 'ant', 'beats', 'dangerous']
2 .  |  B000002H84  |   4.445    |     ['beats', 'tracks', 'know', 'main', 'classic']
3 .  |  B000003C3V  |   4.391    |     ['tha', 'thugs', 'classic', '1999', 'crossroads']
4 .  |  B000001A6X  |   4.384    |     ['wonder', 'life', 'key', 'duke', 'wish']
5 .  |  B0000013GT  |   4.359    |     ['hip', 'hop', 'beats', 'rap', 'big']
6 .  |  B000002AV5  |   4.358    |     ['smooth', 'voice', 'sades', 'ordinary', 'sweetest']
7 .  |  B000003B7Z  |   4.343    |     ['coolio', 'hop', 'hip', 'damn', 'changed']
8 .  |  B000005CEU  |   4.343    |     ['prince', 'production', 'true', 'poetic', 'hip']
9 .  |  B000001A6N  |   4.335    |     ['innervisions', 'life', 'high', 'better', 'visions']
10.  |  B000005Z0G  |   4.333    |     ['pac', 'tupac', '2pac', 'pacs', 'ice']
11.  |  B0000004UM  |   4.331    |     ['necessary', 'means', 'classic', 'true', 'bdp']
12.  |  B0002IQK40  |   4.328    |     ['pick', 'come', 'new', 'power', 'pop']
13.  |  B0000024LN  |   4.327    |     ['muddy', 'classic', 'funk', 'waters', 'redmans']
14.  |  B00004U91U  |   4.312    |     ['girls', 'possible', 'art', 'ani', 'theyre']
15.  |  B0000024IE  |   4.304    |     ['public', 'rap', 'hip', 'black', 'hop']
16.  |  B0033B2I4Y  |   4.299    |     ['beat', 'step', 'does', 'hop', 'hip']
17.  |  B000000053Q  |   4.298    |     ['southern', 'classic', 'south', 'production', 'favorite']
18.  |  B00000J2PJ  |   4.298    |     ['small', 'brings', 'word', 'years', 'sendak']
19.  |  B000001891  |   4.282    |     ['barmy', 'smith', 'punk', 'early', 'smiths']
20.  |  B0015FS8QC  |   4.275    |     ['actually', 'different', 'half', 'metal', 'expect']

추천목록에서 지우고 싶은 음악의 번호를 입력해주세요. (메뉴로 돌아가시려면 -1을 입력해주세요.): 3, 7
```

`

```
---------- 아래의 음악들이 추천 목록에서 제외되었습니다. ----------
-  B000003C3V
-  B000003B7Z
---------- 유사한 음악들의 추천도가 내려갔습니다. ----------
삭제된 음악 :  B000003C3V  연관된 음악 :  B000003C3Q  유사도 :  45.3 % 예측 평점 변화 :  3.997  ->  3.545
                           연관된 음악 :  B0000024LT  유사도 :  28.0 %  예측 평점 변화 :  3.803  ->  3.537
                           연관된 음악 :  B00000055E  유사도 :  20.9 %  예측 평점 변화 :  3.969  ->  3.761
                           연관된 음악 :  B000002WR5  유사도 :  20.4 %  예측 평점 변화 :  4.138  ->  3.926
삭제된 음악 :  B000003B7Z  연관된 음악 :  B00000051K  유사도 :  38.2 % 예측 평점 변화 :  4.073  ->  3.684
                           연관된 음악 :  B000002H8I  유사도 :  29.2 %  예측 평점 변화 :  4.093  ->  3.794
                           연관된 음악 :  B0000028SB  유사도 :  29.1 %  예측 평점 변화 :  4.04  ->  3.746
                           연관된 음악 :  B000003B3T  유사도 :  28.3 %  예측 평점 변화 :  4.091  ->  3.802
                           연관된 음악 :  B000000W2Z  유사도 :  26.2 %  예측 평점 변화 :  3.999  ->  3.737
                           연관된 음악 :  B000002JV1  유사도 :  25.6 %  예측 평점 변화 :  3.593  ->  3.363
                           연관된 음악 :  B0000027S9  유사도 :  25.1 %  예측 평점 변화 :  3.973  ->  3.724
                           연관된 음악 :  B000001EIU  유사도 :  21.4 %  예측 평점 변화 :  4.2  ->  3.975
                           연관된 음악 :  B000003AHB  유사도 :  21.4 %  예측 평점 변화 :  3.945  ->  3.734
                           연관된 음악 :  B0000024L0  유사도 :  20.8 %  예측 평점 변화 :  4.078  ->  3.866
                           연관된 음악 :  B00000050R  유사도 :  20.8 %  예측 평점 변화 :  4.469  ->  4.237

 ----- user mode -----
1. 추천 목록 출력
2. 추천 목록에서 특정 항목 제거하기
3. 인기곡으로부터 추천받기
4. 돌아가기
select :
```

기존 1등 음악

그림 3. 추천 목록에서 특정 음악을 제거하는 과정

```
---------------------------------------- 추천 목록 ----------------------------------------
---------- item ---- expected rate ------------------------ keyword ---------------------
1 .  |  B000002H84  |   4.445    |     ['beats', 'tracks', 'know', 'main', 'classic']
2 .  |  B000001A6X  |   4.384    |     ['wonder', 'life', 'key', 'duke', 'wish']
3 .  |  B0000013GT  |   4.359    |     ['hip', 'hop', 'beats', 'rap', 'big']
4 .  |  B000002AV5  |   4.358    |     ['smooth', 'voice', 'sades', 'ordinary', 'sweetest']
5 .  |  B000005CEU  |   4.343    |     ['prince', 'production', 'true', 'poetic', 'hip']
6 .  |  B000001A6N  |   4.335    |     ['innervisions', 'life', 'high', 'better', 'visions']
7 .  |  B000005Z0G  |   4.333    |     ['pac', 'tupac', '2pac', 'pacs', 'ice']
8 .  |  B0000004UM  |   4.331    |     ['necessary', 'means', 'classic', 'true', 'bdp']
9 .  |  B0002IQK40  |   4.328    |     ['pick', 'come', 'new', 'power', 'pop']
10.  |  B0000024LN  |   4.327    |     ['muddy', 'classic', 'funk', 'waters', 'redmans']
11.  |  B00004U91U  |   4.312    |     ['girls', 'possible', 'art', 'ani', 'theyre']
12.  |  B0000024IE  |   4.304    |     ['public', 'rap', 'hip', 'black', 'hop']
13.  |  B0033B2I4Y  |   4.299    |     ['beat', 'step', 'does', 'hop', 'hip']
14.  |  B000000053Q  |   4.298    |     ['southern', 'classic', 'south', 'production', 'favorite']
15.  |  B00000J2PJ  |   4.298    |     ['small', 'brings', 'word', 'years', 'sendak']
16.  |  B00000189I  |   4.282    |     ['barmy', 'smith', 'punk', 'early', 'smiths']
17.  |  B0015FS8QC  |   4.275    |     ['actually', 'different', 'half', 'metal', 'expect']
18.  |  B000000W7O  |   4.270    |     ['real', 'rap', 'gangsta', 'face', 'beats']
19.  |  B000B8I940  |   4.270    |     ['years', 'gold', 'ralph', 'set', 'right']
20.  |  B000001E44  |   4.258    |     ['classic', 'funk', 'stuff', 'horns', 'wild']
```

그림 4. 특정 음악을 추천 목록에서 제거한 뒤 변경된 추천 목록

`

특정 음악을 추천 목록에서 제거한 뒤 추천 목록을 다시 보면 제거했을 때 나온 메시지와 같이 3등과 7등이던 음악 'B000003C3V', 'B000003B7Z', 그리고 제거된 음악 'B000003B7Z'와 유사하여 평점이 낮아진 음악 'B00000050R'이 추천 목록에서 사라진 것을 확인할 수 있다. 그에 따라 기존에 이들보다 아래에 있었던 음악들의 순위가 높아진 것 역시 변경된 추천 목록을 통해 확인할 수 있다. 사용자 모드 모드 나갈 때 예측 평점 행렬은 파일로 저장되며, 따라서 프로그램을 종료해도 평점 변경 내용은 유지된다.

마지막으로, 인기 차트에 올라와 있는 음악들 중 마음에 드는 음악이 있으면 그 음악과 관련된 다른 음악들을 추천해주고, 추천한 음악들의 예측 평점을 올려주는 기능을 수행할 수 있다. 예측 평점은 최대 5점까지 증가할 수 있다.

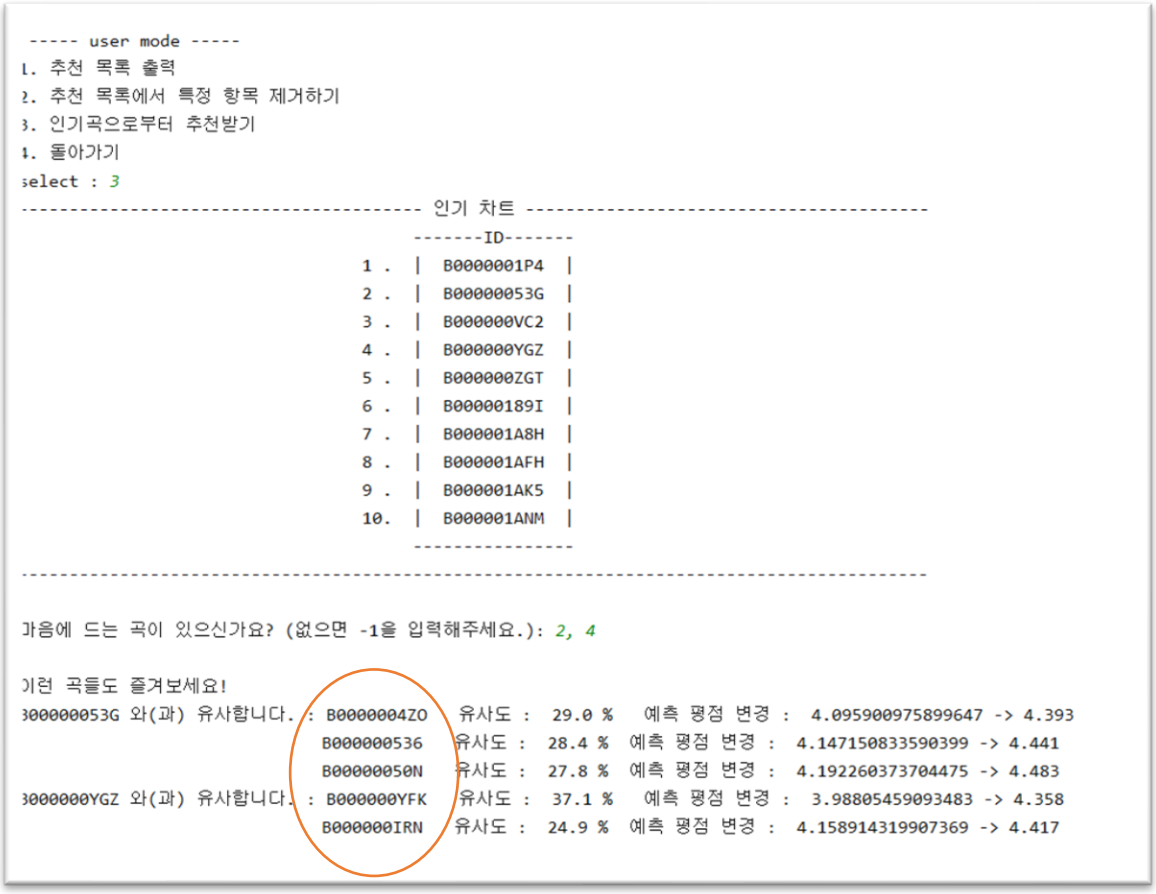


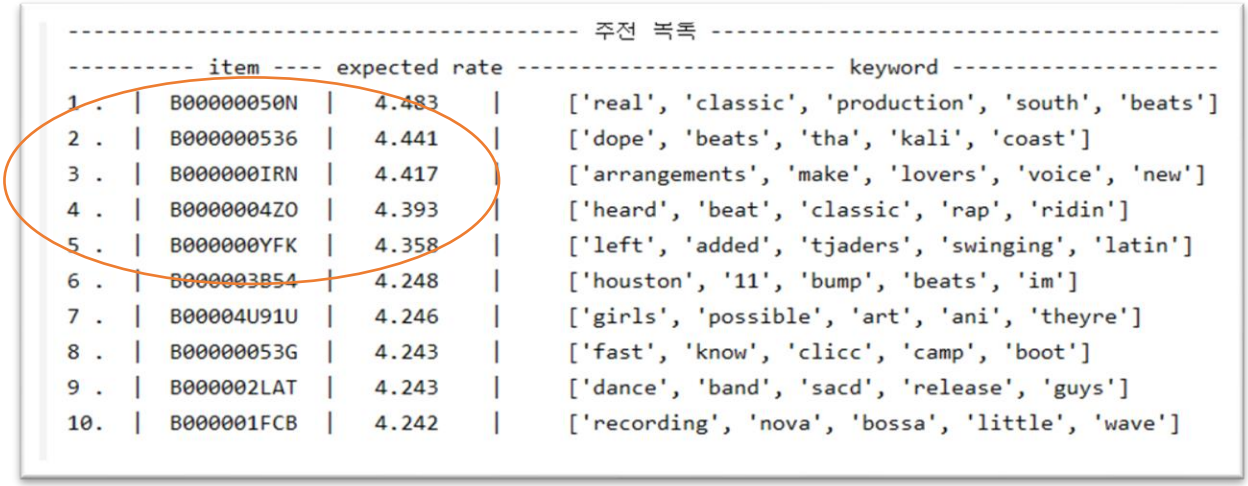
그림 5. 인기 차트에서 특정 음악을 통해 관련된 음악을 추천 받는 모습

변경된 예측 평점은 다음과 같이 추천 목록에 반영된다.

`

추천 목록

	item	expected rate	keyword
1 .	B000003B54	4.248	['houston', '11', 'bump', 'beats', 'im']
2 .	B00004U91U	4.246	['girls', 'possible', 'art', 'ani', 'theyre']
3 .	B00000053G	4.243	['fast', 'know', 'clicc', 'camp', 'boot']
4 .	B000002LAT	4.243	['dance', 'band', 'sacd', 'release', 'guys']
5 .	B000001FCB	4.242	['recording', 'nova', 'bossa', 'little', 'wave']
6 .	B00CDX176I	4.241	['relaxation', 'work', 'hard', 'meditation', 'sleep']
7 .	B000002LJF	4.240	['hip', 'rap', 'caps', 'played', 'weapon']
8 .	B000008BFI	4.240	['domestically', 'madness', 'worth', 'getting', 'effort']
9 .	B00000J2PJ	4.240	['small', 'brings', 'word', 'years', 'sendak']
10.	B00005J9U8	4.240	['stand', 'classic', 'game', 'tight', 'heard']



주전 목록

	item	expected rate	keyword
1 .	B00000050N	4.483	['real', 'classic', 'production', 'south', 'beats']
2 .	B000000536	4.441	['dope', 'beats', 'tha', 'kali', 'coast']
3 .	B000000IRN	4.417	['arrangements', 'make', 'lovers', 'voice', 'new']
4 .	B0000004ZO	4.393	['heard', 'beat', 'classic', 'rap', 'ridin']
5 .	B000000YFK	4.358	['left', 'added', 'tjaders', 'swinging', 'latin']
6 .	B000003B54	4.248	['houston', '11', 'bump', 'beats', 'im']
7 .	B00004U91U	4.246	['girls', 'possible', 'art', 'ani', 'theyre']
8 .	B00000053G	4.243	['fast', 'know', 'clicc', 'camp', 'boot']
9 .	B000002LAT	4.243	['dance', 'band', 'sacd', 'release', 'guys']
10.	B000001FCB	4.242	['recording', 'nova', 'bossa', 'little', 'wave']

예측 평점 행렬은 사용자 메뉴에서 나갈 때 파일로 저장된다. 따라서 이러한 변경점들은 프로그램을 종료해도 계속 유지된다.

5. 어려웠던 점

이 프로젝트를 진행하면서 해결하기 어려웠던 많은 문제점들이 있었다. 발생한 문제점들과 이에 대한 우리 나름대로의 해결 방법을 정리했다.

`

1. 예측 평점의 평점 범위 초과 문제

우리의 추천 시스템은 기본적으로 matrix factorization 기법을 사용하기 때문에 오차로 인해 추천 목록에서 보여주는 예측 평점 값이 평점 범위인 1~5점 사이를 벗어날 수 있었다. 이 문제점을 해결하기 위해 예측 평점들을 대상으로 리스케일링 과정을 수행했다. 리스케일링은 아래와 같은 과정을 거친다.

1. 예측 평점들 중 최댓값과 최솟값을 찾고, 그 차이를 변수 denominator로 저장한다.
2. 각 예측 평점들에서 최솟값을 뺀 뒤 변수 denominator로 나눈다. 이를 통해 예측 평점의 범위를 0에서 1 사이로 맞출 수 있다.
3. 결과값에 4를 곱한 뒤 1을 더해서 범위를 1에서 5 사이로 맞춘다.

2. 추천목록의 순서가 매 학습마다 바뀌는 문제

우리가 만든 추천 시스템은 매번 학습을 진행할 때마다 음악 추천 순위가 급격하게 변한다는 치명적인 문제를 가지고 있었다. 이를 해결할 수 있는 방법을 교수님께 물어본 결과 train set의 개수를 늘리면 순위 변화를 줄일 수 있을 것이라는 말을 들었고, test set을 기존의 5000개에서 1000개로 줄이고 4000개를 train set에 추가한 결과 이전보다는 훨씬 더 개선된 결과를 얻을 수 있었다.

3. 각 음악과 관련된 키워드 추출 관련 문제

우리는 각 음악에 관련된 키워드를 표시하기 위해 평점 데이터와 같이 저장되어 있는 유저들의 리뷰를 이용했다. 리뷰를 아이템별로 모아 그 안에서 TF-IDF 기법을 통해 리뷰에서 자주 등장하는 순으로 상위 5개의 키워드를 아이템과 같이 표시했는데, 키워드에 그 음악의 고유한 특성이 아닌 보편적인 단어가 들어가는 문제점을 발견할 수 있었다. The나 he, she와 같은 단어들은 python의 라이브러리인 sklearn에서 제공하는 stopword 리스트를 사용해 제거할 수 있었지만, 음악 리뷰로부터 단어를 뽑아냈기 때문에 song이나 album, singer와 같은 단어들은 제거할 수 없었다. 이를 해결하기 위해 음악 리뷰에서 자주 등장하는 song, album, singer와 같은 단어들을 extend 리스트에 추가하고, 이를 stopword 리스트와 합쳐서 TF-IDF 수행 시에 해당 단어들을 제외하고 계산할 수 있게 하였다. 또한 DF값이 2 이상인 경우에만 TF-IDF값을 계산하게 하여 하나의 문서에만 자주 등장하는 단어들은 배제할 수 있도록 하였다.

`