Playlist Continuation

회귀분석팀 권남택 윤주희 진효주 한유진 황유나

[CONTENTS]

1

1주차 피드백

- 피드백

2

문제상황

- NDCG란?
- 분석 방향

3

데이터 탐색

- MF (행렬 분해)
- CF (협업 필터링)
- 추천시스템 구현

4

한계 및 의의

- 한계
- 의의

02. 문제 상황

20:00 셔플재생
01. 1주차 피드백
문제 상황
03. 모델링
04. 한계 및 의의

- 모델의 예측 정확도 평가 지표: nDCG

채점 방법: score = 평균 nDCG(예측한 곡) * 0.85 + 평균 nDCG(예측한 태그) * 0.15

"nDCG를 계산할 때 정답 안에서의 순서는 무시됩니다!"

ⓘ 하지만, 일반적인 추천 시스템에서는

순서	사용자가 좋아할 만한 컨텐츠를 상단에 배치 더 많은 사용자를 유인할 수 있는 키워드를 앞부분에 배치
개인화	사용자의 선호 컨텐츠를 분석하여 이와 유사한 컨텐츠를 추천

02. 문제 상황

20:00 셔플재생
01. 1주차 피드백
문제 상황
03. 모델링
04. 한계 및 의의

- 모델의 예측 정확도 평가 지표: nDCG

$$Recommendations\ Order = [2, 3, 3, 1, 2]$$

$$Ideal\ Order = [3, 3, 2, 2, 1]$$

*상단부터 예측치, 정답

$$DCG = \sum_{i=1}^{n} \frac{relevance_i}{log_2(i+1)}$$

*relevance: 가중치

가중치가 높은 항목이 먼저 배치될 때 더 큰 값을 갖게 되는 형태

항목 배치 순서가 중요하다!

02. 문제 상황

20:00 셔플재생

01. 1주차 피드백

문제 상황

03. 모델링

04. 한계 및 의의

- 데이터의 빈칸 처리

태그	playlist id	playlist name	수록 곡 id	좋아요 개수	업데이트 일자
[]	118598		[373313, 151080, 275346, 696876, 165237, 52593...	1675	2019-05-27 14:14:33.000
[]	131447	앨리스테이블	[]	1	2014-07-16 15:24:24.000
[]	51464		[529437, 516103, 360067, 705713, 226062, 37089...	62	2008-06-21 23:26:22.00
['잔잔한']	101722		[75842, 26083, 244183, 684715, 500593, 508608,...	17	2015-12-17 14:06:05.000
['어머니 ... 들', '아빠', '... 속', '위로받고싶을 때']	122127		[450275, 487671, 561031, 663944, 628672, 59121...	10	2020-04-16 21:35:44.000

③ 태그가 적은 경우

03. 모델링

MF(행렬 분해) CF(협업 필터링) 추천시스템 구현

20:00 셔플재생

01. 1주차 피드백

02. 문제상황

모델링

04. 한계 및 의의

- Matrix Factorization

Loss Function

예측한 평점과 실제 평점의 차이를 나타내는 수식

$$\min_{x^*,\ y^*} \sum_{u,i} (r_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

^ 차이를 최소화 ^ 과적합 방지

Optimization

① Stochastic Gradient Descent
②Alternating Least Squares (더 많이 사용)

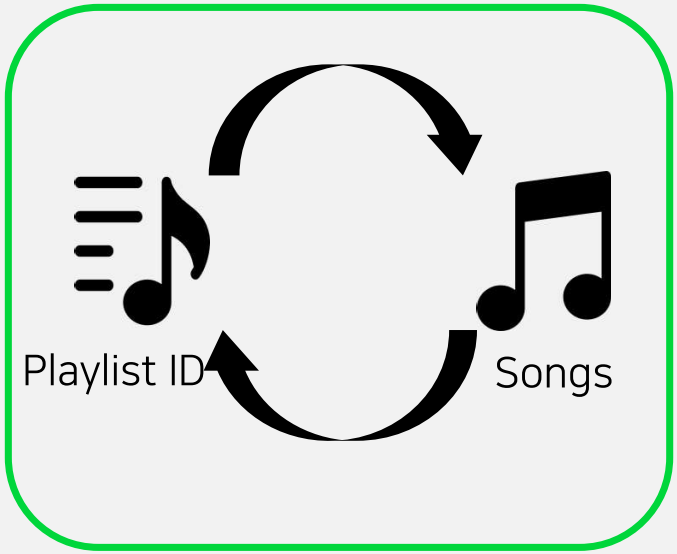
→ 우리가 사용할 방법

20:00
셔플재생
01. 1주차 피드백
02. 문제상황
모델링
04. 한계 및 의의

03. 모델링

- Matrix Factorization

ALS 모델 파라미터의 손실을 줄이는 알고리즘 이용



플레이리스트 id와 Song을 번갈아 고정시키면서 최적화

$$\min_{x^*,y^*} \sum_{u,i} (r_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

① X를 고정해서 최적화

$$\min_{x^*,y^*} \sum_{u,i} (r_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right)$$

② Y를 고정해서 최적화

03. 모델링

20:00 셔플재생
01. 1주차 피드백
02. 문제상황
모델링
04. 한계 및 의의

- Factorization Machine

	User				Item				Categories				History				Quantity	
x^1	1	0	0	...	1	0	0	...	1	0	1	...	1	0	1	...	2	y^1
x^2	1	0	0	...	0	1	0	...	0	2	1	...	0	0	1	...	4	y^2
x^3	0	1	0	...	1	0	0	...	3	0	14	...	1	0	0	...	5	y^3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x^n	0	0	0	...	0	0	1	...	0	1	5	...	0	0	1	...	1	y^n

User, Item 이외에 존재하는 다양한 feature들을 적용하여 학습 시키는 모델
변수들 간의 interaction을 새로운 파라미터 d로 모델링 할 수 있음

Optimization

① Stochastic Gradient Descent (SGD)
② Alternating Least Squares (ALS)
③ Markov Chain Monte Carlo (MCMC)

3가지 모두 이용

03. 모델링

- Word2vec – 개념 및 활용

20:00
01. 1주차 피드백
02. 문제상황
모델링
04. 한계 및 의의

태그1 태그2 태그3 태그4 태그5 태그6 태그7 태그8 태그9 태그10

Word2vec 이용해서 유사한 태그들 중 '코사인 유사도' 가 0.6이상인 태그 추출!

플레이리스트에 있는 태그
n= 각기 다름

But… 태그와 별로 유사하지가 않다 ?

우리가 예측할 태그
비어 있는 곳을 skip-gram으로!

	tag	similiarit
4	[잔잔한]	[잔잔함, 추운날, 해외차트, 마틀, 해외음악, 만화주제가]
5	[드라이브]	[봄노래, 멜랑꼴리, 나이, 모음집, 집에가는길, 길거리, 초여름]
6	[힐링]	[작곡가, 소다기, 8월, 고요, 집에서, 스타일리시, 나만알고싶은노래, 락밴드]
7	[사랑]	소녀, COOL, 추운, 후유증, 헬스음악, 편집샵, 평온]
8	[새벽]	창모, 프로듀싱, 5월, 뜨거운, 텐션업, 봄소풍]
9	[밤]	[크로스핏, 헬스장, 레이디가가, 인생노래, 불토, 달리기, 밤에, 클래식음악]
10	[카페]	[유니크한, 슬로우잼, 새로운시작, 박효신]
11	[추억]	[비오는_날, 질리지않는, 위너, 디저트, 밤산책, Labelpick]

모델링할 데이터와 변수를 바꿔가며…

Word2vec 만으로
최적의 추천을 해보자!

03. 모델링

20:00 셔플재생
01. 1주차 피드백
02. 문제상황
모델링
04. 한계 및 의의

- AutoEncoder : 행렬 분해와 차이점

행렬 분해는 행렬간의 선형적인 연산을 통한 Latent Factor 기반의 연산

→ 간단하면서 강력하지만 한계가 존재!

03. 모델링

20:00 셔플재생
01. 1주차 피드백
02. 문제상황
모델링
04. 한계 및 의의

- AutoEncoder : 더 좋은 모델은?

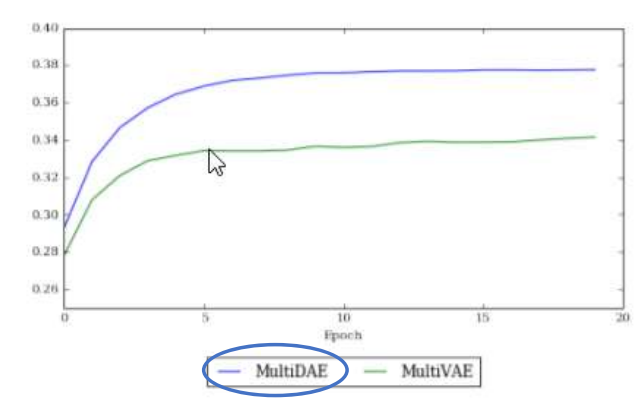


Figure 2: NDCG@100 of Mult-DAE and Mult-VAE on validation data

Don't Stop the Music : Playlist Continuation with Autoencoders

Table 3: Performance of different recommender systems (RecSys) measured as recall (r) and mean reciprocal rank (mrr).

Dataset	RecSys	r/mrr @1	recall @5	recall @20	mrr @5	mrr @20
AotM	unigram	0.001	0.004	0.016	0.002	0.003
	bigram	0.015	0.027	0.031	0.020	0.020
	trigram	0.015	0.015	0.015	0.015	0.015
	kNNi20	0.027	0.044	0.062	0.033	0.035
	kNNc50	0.018	0.024	0.036	0.020	0.022
	AE1	**0.028**	**0.048**	0.076	**0.035**	0.038
	AE2	**0.028**	0.045	0.073	**0.035**	**0.039**
	AE3	0.027	0.045	**0.080**	0.034	0.037
	AE4	0.027	0.045	0.069	0.034	0.036

Autoencoders for Next-Track-Recommendation

VAE(Variational AE)와 같이 더 복잡한 모델을 쓰지 않아도
Playlist Continuation에서 AE도 성능이 좋다는 논문 확인!

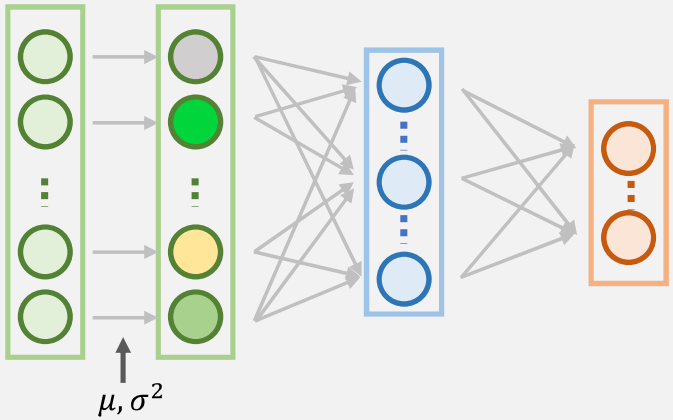
사실 시간도 없었다...

03. 모델링

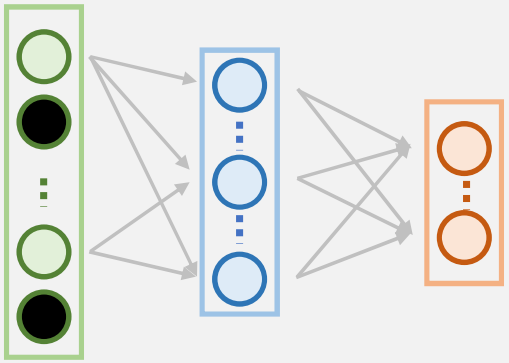
20:00 셔플재생
01. 1주차 피드백
02. 문제상황
모델링
04. 한계 및 의의

- Denoising AutoEncoder

1. Gaussian Noise input



2. Dropout input



우리 상황에 맞게 Dropout input 방식 채택

03. 모델링

20:00
셔플재생
01. 1주차 피드백
02. 문제상황
모델링
04. 한계 및 의의

- KNN 기반 추천시스템

```
self.train_id = train["id"]
self.train_songs = train["songs"]
self.train_tags = train["tags"]
del train

self.val_id = val["id"]
self.val_songs = val["songs"]
self.val_tags = val["tags"]
del val
```

```
from knn import KNN

k = 100
alpha = 0.5
beta = 0.5
sim_songs = "cos"
sim_tags = "cos"
sim_normalize = False

pred = KNN(k=k, alpha=alpha, beta=beta, #
           sim_songs=sim_songs, sim_tags=sim_tags, sim_normaliz
           train=train, val=val, verbose=True, version_check=T
# print(pred)
```

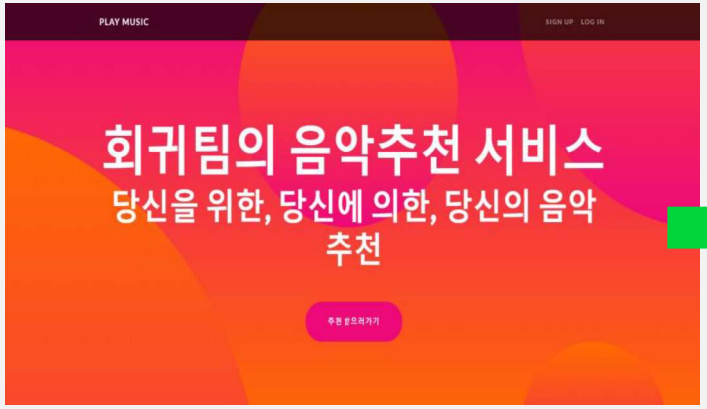
Train, Validation의 tag, song에 대해
K=100, 유사도는 코사인 유사도로 측정하였다.

03. 모델링

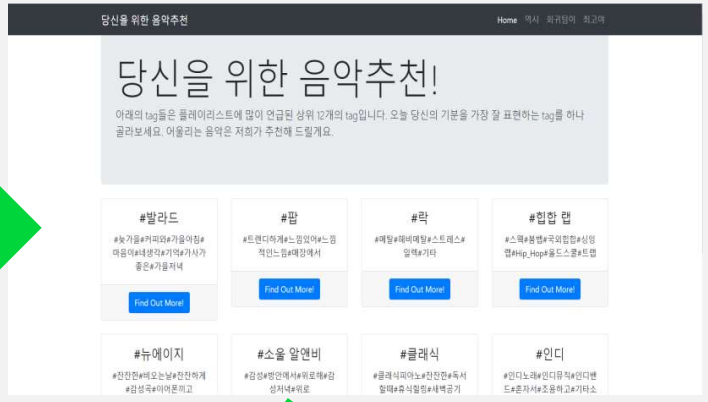
MF(행렬 분해) CF(협업 필터링) 추천시스템 구현

20:00 셔플재생
01. 1주차 피드백
02. 문제상황
모델링
04. 한계 및 의의

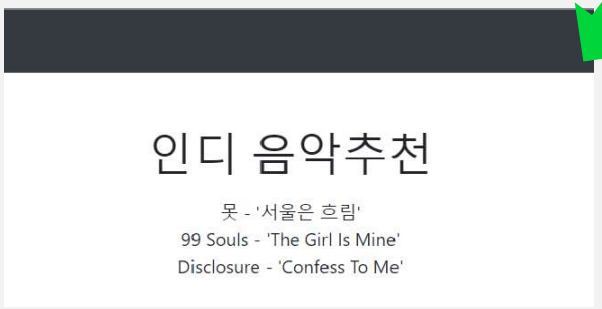
- KNN 알고리즘 기반 추천시스템 구현



① 추천 시스템 메인 페이지



② 장르 태그 선택



③ 알고리즘 기반 노래 추천

04. 한계 및 의의

한계 의의

20:00 셔플재생

01. 1주차 피드백

02. 문제상황

03. 모델링

한계 및 의의

의의

- 다양한 모델을 살펴보고, 직접 실행해보는 값진 경험 (MF, FM, KNN, w2v, auto-encoder…)
- 파이썬과 R 둘 다 이용해서 모델을 다뤄보았다!
- 다루기 어려운 데이터도 포기하지 않고 할 수 있다!!!