



토픽 모델링을 이용한 음악 추천 시스템

Music Recommender System using Topic Modeling

홍익대학교 일반대학원 산업공학과

데이터마이닝 Lab

박대한

목차

1. 서론

2. 관련 연구

- ❖ Recommendation Algorithm
- ❖ Topic Modeling

3. 제안 기법

4. 실험 및 성능 평가

- ❖ 데이터 준비
- ❖ 실험 결과

5. 결론 및 향후 연구과제

6. 참고 문헌

● 연구 배경

- ❖ 음악 도메인은 다른 도메인보다 Explicit 피드백 데이터 획득이 어려움 [1]
 - ◆ Item의 소비 시간이 짧아 사용자들이 피드백 데이터를 제공하지 않음
 - Explicit 피드백 데이터 : Rating, Thumb up and down
 - ◆ 피드백 데이터가 덜 필요한 Content 기반의 음악 추천이 많이 연구되어짐
 - ◆ 오디오 신호를 이용한 추천은 음악의 복잡한 특성을 충분히 반영하지 못함 [2]
- ❖ Tag는 텍스트 기반의 키워드로서 곡에 대한 다양한 사용자의 의견을 표현
 - ◆ 장르 : "rock", 분위기 : "sad", 아티스트 특성 : "female voice", 경험 : "seen live"
 - ◆ 오디오 신호가 표현할 수 없는 사용자의 개인적인 반응에 대한 정보를 담고 있다는 점에서 매우 유용한 정보 [3]
- ❖ Tag는 다양한 정보를 담고 있지만 한계점이 존재
 - ◆ Tag는 Noisy하고, 동의어, 다의어에 대한 표현이 불가
 - ◆ 이를 해결하기 위한 다양한 Topic Modeling 기법들이 존재

● 연구 목적

- ❖ Tag를 이용한 음악 추천 시스템의 추천 정확도 향상
- ❖ Tag로 부터 추출된 Topic을 이용하여 음악의 특성을 해석

● 연구 방법

- ❖ Topic Modeling 기법을 적용한 음악 추천 시스템
 - ◆ Tag에 Latent Dirichlet Allocation(LDA)을 적용하여 Topic을 생성
 - ◆ 사용자의 청취 이력과 음악의 Topic 정보를 이용하는 Hybrid 추천 방법을 적용
 - Item-based Collaborative Filtering : 사용자 청취 이력을 이용
 - Topic-based Content-based Filtering : 사용자 청취 이력과 Topic 정보 이용
 - 두 알고리즘의 Score를 Hybrid하여 최종 Score 예측

● Recommendation Algorithm

- ❖ Content-based Filtering (CBF) [4]
- ❖ Collaborative Filtering (CF)
 - ◆ User-based CF [5]
 - ◆ Item-based CF [6]
- ❖ Knowledge-based Recommendation [7]
- ❖ Demographic Recommendation [8]
- ❖ Hybrid Recommendation [7]
 - ◆ Weighted 방법 [9]
 - CBF와 CF의 예측 Score를 결합하므로써 두 알고리즘의 단점을 보완

< 알고리즘 장단점 비교 >

	장점	단점
Content-based Filtering	<ul style="list-style-type: none"> • New Item problem 해결 • Item의 Content를 제시함으로써 추천에 대한 설명 가능 	<ul style="list-style-type: none"> • 일부 도메인에서는 속성 추출이 어려움 • 다양한 Item이 추천되지 않음
Collaborative Filtering	<ul style="list-style-type: none"> • 다양한 Item을 추천 • 도메인과 상관없이 적용 가능 	<ul style="list-style-type: none"> • New User 문제 • New Item 문제 • 인기가 많은 Item 위주로 추천
Knowledge-based Recommendation	<ul style="list-style-type: none"> • 사용자의 요구에 부합하는 Item 탐색 가능 • 피드백 데이터가 필요 없음 	<ul style="list-style-type: none"> • 사전에 Knowledge 구축 필요
Demographic Recommendation	<ul style="list-style-type: none"> • 피드백 데이터가 필요 없음 	<ul style="list-style-type: none"> • 사용자의 Demographic 정보 수집 필요

● Recommendation Algorithm

❖ 음악 관련 연구

◆ CF [10]

- Rating 데이터에 User-based CF를 적용한 최초의 음악 추천 시스템

◆ CBF [11]

- 곡으로부터 템포, 음조, 리듬, 길이, 톤, 음색을 추출한 뒤 Decision Tree 적용

◆ CBF [12]

- 곡의 Latent factor을 추출하기 위하여 오디오 신호에 Deep convolutional neural networks 적용

◆ Hybrid Recommendation [13]

- Cold Start를 해결하기 위해 사용자의 청취 횟수와 Tag 점수를 가중 평균하여 Rating을 새로 계산한 뒤 User-based CF 적용

※ 본 연구와의 차이점

- Tag가 아닌 Topic Modeling을 통해 추출된 Topic을 이용
- Hybrid 방식으로 두 알고리즘의 Score값을 가중 평균하는 Weighted 방법 적용

● Recommendation Algorithm

❖ 음악 관련 연구

◆ LDA를 이용한 CF [13]

- 곡의 Tag에 LDA를 적용하여 Topic 추출
- Topic 확률 값과 청취 여부를 이용하여 Score 계산

※ 본 연구와의 차이점

- Topic을 Item의 특성으로 CBF에서 적용 후, Score를 CF와 CBF의 Hybrid하는 방식으로 계산

◆ LDA를 이용한 Hybrid Recommendation [14]

- 특성이 다른 CF들을 결합하여 다양하고 참신한 Artist를 추천
 - 사용자가 들은 곡을 Artist 단위로 합계를 내어 Rating 데이터 구성
 - LDA를 적용하여 여러 Artist들의 모음으로 구성된 Topic을 추출
 - 곡의 Topic 확률 값과 청취 여부를 이용하여 Item-based CF를 적용

▪ ※ 본 연구와의 차이점

- Artist 단위가 아닌 곡 단위에 대하여 LDA를 적용

● Topic Modeling

- ❖ 단어로 구성되어있는 문서에서 Topic를 추출해 내는 기법
 - ◆ 단어와 Latent한 주제 사이의 상관 관계를 추출 [16]
 - ◆ Topic : 단어의 모음들로 표현되는 주제

- ❖ 종류
 - ◆ Latent Semantic Analysis (LSA) [17]
 - ◆ Probabilistic Latent Semantic Analysis (PLSA) [18]
 - ◆ Latent Dirichlet Allocation (LDA) [19]

< Topic Modeling 장단점 비교 >

	장점	단점
Latent Semantic Analysis (LSA)	<ul style="list-style-type: none"> 차원 축소로 인한 Noise 제거 동义词 문제 해결 	<ul style="list-style-type: none"> 다의어 문제 문서를 단어의 vector로 표현하기 때문에 문맥이 고려되지 않음 (Bag of Words) 결과의 해석이 어려움
Probabilistic Latent Semantic Analysis (PLSA)	<ul style="list-style-type: none"> 다의어 문제 해결 	<ul style="list-style-type: none"> 학습되지 않은 문서에 대한 일반화 불가 학습되는 파라미터 수가 문서의 수에 비례 Overfitting 문제
Latent Dirichlet Allocation (LDA)	<ul style="list-style-type: none"> 학습되지 않은 문서에 대한 일반화 가능 Overfitting 문제 해결 	<ul style="list-style-type: none"> 파라미터 값에 대한 사전 설정 필요

● Topic Modeling

❖ Latent Dirichlet Allocation

◆ 문서가 생성되는 과정을 확률적(stochastic)으로 모델링

- 문서로부터 Topic이 정해지고, 해당 Topic으로부터 단어가 정해짐
- 문서의 Topic 확률과 Topic의 단어 확률은 사전 확률(Prior distribution)로 가정
 - 사전 확률 분포 : 디리클레 분포(Dirichlet distribution)

◆ 생성 과정(Generative Process)

1. For each topic k :

- Draw a distribution over words : $\phi_k \sim Dir(\beta)$, $k \in \{1, \dots, K\}$
 - $Dir(\cdot)$: uniform Dirichlet distribution
 - β : parameter

2. For each document d :

a) Draw a vector of topic proportions : $\theta_d \sim Dir(\alpha)$, $d \in \{1, \dots, D\}$

b) For each word v :

- (i) Draw a topic assignment : $z_{d,v} \sim Mult(\theta_d)$, $d \in \{1, \dots, D\}$, $v \in \{1, \dots, V\}$
- (ii) Draw a word : $w_{d,v} \sim Mult(\phi_{z_{d,v}})$, $d \in \{1, \dots, D\}$, $v \in \{1, \dots, V\}$

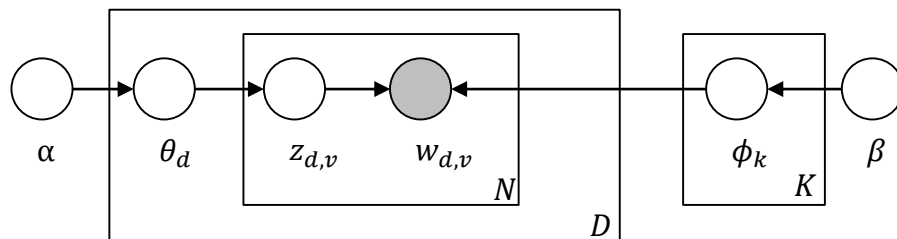
$$p(w, z, \theta, \phi | \alpha, \beta) = p(\theta | \alpha) p(z | \theta) p(\phi | \beta) p(w | z, \phi)$$

$$p(\theta | \alpha) = \prod_{d=1}^D \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K \theta_k^{\alpha_k - 1}$$

$$p(z | \theta) = \prod_{d=1}^D \prod_{k=1}^K \theta_{d,k}^{n_{d,k}}$$

$$p(\phi | \beta) = \prod_{k=1}^K \frac{\Gamma(\sum_{v=1}^V \beta_{k,v})}{\prod_{v=1}^V \Gamma(\beta_{k,v})} \prod_{v=1}^V \phi_{k,v}^{\beta_{k,v} - 1}$$

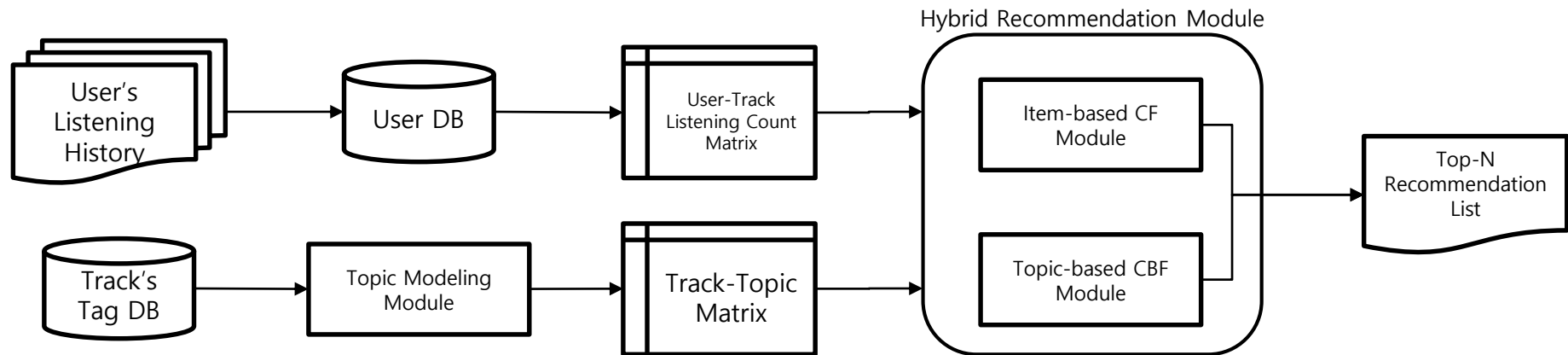
$$p(w | z, \phi) = \prod_{k=1}^K \prod_{v=1}^V \phi_{k,v}^{n_{k,v}}$$



< LDA plate notation >

● Overview

- ❖ (1) Topic Modeling Module
- ❖ (2) Hybrid Recommendation Module
 - ◆ Topic-based CBF Module
 - ◆ Item-based CF Module



< 제안 기법 개요 >

● Topic Modeling Module

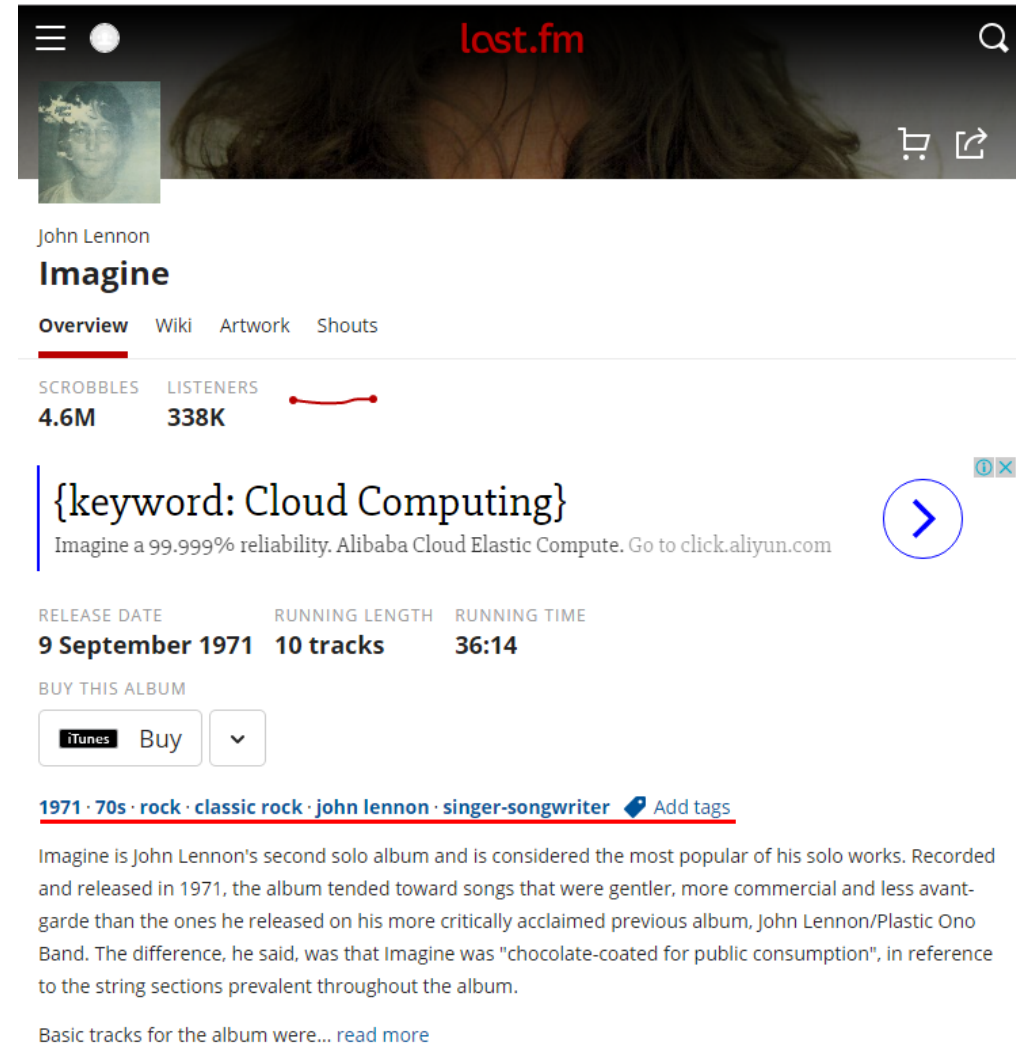
❖ Tag 수집

◆ Last.fm

- 인터넷 라디오를 제공하는 영국의 음악 소셜 네트워크 웹사이트

◆ Last.fm Tag 유형 [20]

- 장르 : 60%
- 분위기/의견/선호 : 20%
- 테마/상황 : 5%



last.fm

John Lennon

Imagine

Overview Wiki Artwork Shouts

SCROBBLES 4.6M LISTENERS 338K

{keyword: Cloud Computing}

Imagine a 99.999% reliability. Alibaba Cloud Elastic Compute. Go to click.aliyun.com

RELEASE DATE 9 September 1971 RUNNING LENGTH 10 tracks RUNNING TIME 36:14

BUY THIS ALBUM

iTunes Buy

1971 · 70s · rock · classic rock · john lennon · singer-songwriter Add tags

Imagine is John Lennon's second solo album and is considered the most popular of his solo works. Recorded and released in 1971, the album tended toward songs that were gentler, more commercial and less avant-garde than the ones he released on his more critically acclaimed previous album, John Lennon/Plastic Ono Band. The difference, he said, was that Imagine was "chocolate-coated for public consumption", in reference to the string sections prevalent throughout the album.

Basic tracks for the album were... [read more](#)

< Last.fm 웹사이트 >

● Topic Modeling Module

❖ Tag 수집 (Cont.)

- ◆ Last.fm API를 통해 사용자가 청취한 음악의 Tag와 Tag Score를 수집
 - Tag Score : 곡에 달린 Tag 횟수를 정규화한 0~100 사이의 값

< Last.fm Tag Example >

(a) John lennon - Imagine

No	Tag	Score	No	Tag	Score
1	classic rock	100	11	peace	13
2	70s	57	12	beautiful	13
3	john lennon	56	13	the beatles	12
4	rock	52	14	classic	11
5	british	42	15	favorites	9
6	pop	22	16	ballad	9
7	singer-songwriter	21	17	love	8
8	piano	17	18	beatles	8
9	oldies	16	19	lennon	7
10	imagine	13	20	male vocalists	7

(b) Coldplay – The scientist

No	Tag	Score	No	Tag	Score
1	coldplay	100	11	sad	20
2	rock	92	12	pop	18
3	alternative	82	13	beautiful	18
4	britpop	74	14	melancholy	15
5	british	54	15	00s	10
6	alternative rock	37	16	indie rock	10
7	indie	24	17	chillout	10
8	mellow	21	18	melancholic	9
9	piano	21	19	favorites	9
10	love	20	20	the scientist	9

● Topic Modeling Module

❖ Tag 전처리 [21][22]

1. 소문자화

2. Stemming

- Porter Algorithm [23]

3. 선호 관련 Tag 제거

- 음악에 대한 개인의 선호도는 사용자들의 청취횟수로부터 반영되므로 판단하는 의미의 Tag는 불필요
- ex) "bad", "good", "great", "7 of 10 stars", "5 stars"...

4. 특수 기호 통일

- &, n \rightarrow and
 - ex) "rock n roll", "rock & roll", "rock n' roll" \rightarrow "rock and roll"
- 's년도 \rightarrow 숫자
 - ex) 1960's, 60's, 60s \rightarrow 1960

● Topic Modeling Module

❖ Tag 전처리 (Cont.)

5. 수작업으로 Tag 수정

6. 중복되는 Tag 및 Tag의 Score 통합

7. 4개 미만의 곡에 달린 Tag 제거

Tag	수정된 Tag	Tag	수정된 Tag	Tag	수정된 Tag
a lot like love	love	feeling sad	sad	acoustic song versions	acoustic
a song for lovers		make me feel sad		acoustic songs	
about love		makes me sad		acoustic version	
absolute love		mood sad and doomed		acoustic versions	
all about love		mood: sad		acoustic-y	
amazing songs about love		mood: sad - slightly		acoustically acoustical	
feeling in love		sad		acoustique	
just love		sad melody		good acoustic	
make love		sad memories			
mood for love		sad mood			
music for lovers		sad music			
music to make love on		sad sad sad			
pretty love		sadness			
songs about a lover		sads			
songs for love		songs for sad moods			
wanting love					

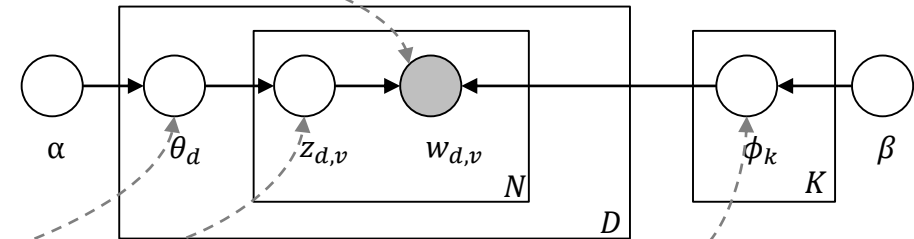
< 수작업 Tag 수정 Example >

● Topic Modeling Module

❖ Topic Modeling

- ◆ Topic Modeling으로 LDA를 적용
- ◆ Input : 곡의 Tag Score
- ◆ Output : 곡의 Topic 분포, Topic의 Tag 분포

	Tag1	Tag2	Tag3	Tag4	Tag5
Track1	0	0	30	0	80
Track2	100	10	0	0	0
Track3	0	0	0	20	100
Track4	0	40	50	0	0



	Topic1	Topic2	Topic3
Track1	0.273	0.726	0.001
Track2	0.001	0.001	0.998
Track3	0.001	0.998	0.001
Track4	0.998	0.001	0.001

	Tag1	Tag2	Tag3	Tag4	Tag5
Track1	-	-	Topic2	-	Topic1
Track2	Topic3	Topic3	-	-	-
Track3	-	-	-	Topic1	Topic2
Track4	-	Topic2	Topic1	-	-

	Tag1	Tag2	Tag3	Tag4	Tag5
Topic1	0	0	0	0.100	0.900
Topic2	0	0.333	0.666	0	0
Topic3	0.909	0.091	0	0	0

< LDA Example >

● Rating Normalization

- ❖ 청취 횟수는 Implicit Rating으로 좋고 나쁨의 기준이 없음 [24]
- ❖ 사용자마다 곡의 평균 청취 횟수가 다름
 - ◆ ex) A 사용자 : 1곡, B 사용자 : 20곡
- ❖ L2-Normalization
 - ◆ 청취 횟수가 많은 곡과 적은 곡의 차이를 살리고자 함

$$\vec{c}_u = (c_{u1}, c_{u2}, \dots, c_{uN})$$

$$\vec{r}_u = \frac{\vec{c}_u}{|\vec{c}_u|}$$

\vec{c} : 청취 횟수 벡터
 u : 추천대상이 되는 음악
 N : 총 음악의 수

	Track1	Track2	Track3	Track4
User1	5	5	0	5
User2	15	10	0	3
User3	1	0	30	2
User4	1	50	0	0



	Track1	Track2	Track3	Track4
User1	0.58	0.58	0	0.58
User2	0.82	0.55	0	0.16
User3	0.03	0	1.00	0.07
User4	0.02	1.00	0	0

<Normalized Rating Example>

● Topic-based CBF Module

1. Topic Modeling Module을 통해 곡의 Topic을 생성

- ◆ Item의 특성으로 Topic을 이용

2. k-NN 알고리즘 적용

- ◆ 1) 유사도 계산

- Topic 확률 값들로부터 곡들간의 유사도를 계산
- 유사도 종류 : Cosine 유사도

- ◆ 2) 이웃 선정

- 유사도가 큰 상위 k 개의 음악 선택

- ◆ 3) Score 계산

- 이웃의 Normalized Rating을 유사도로 가중 평균하는 Score 계산

$$\vec{t} = (\theta_1, \theta_2, \dots, \theta_k)$$

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|}$$

$$score_{t,u,t} = \frac{\sum_{i=1}^k r_{u,t} \times sim(\vec{t}, \vec{t}_i)}{\sum_{i=1}^k |sim(\vec{t}, \vec{t}_i)|}$$

t : 추천대상이 되는 음악
 θ : 해당 토픽에 속할 확률 값
 k : 이웃 수
 i : 이웃 음악

	Topic1	Topic2	Topic3	User1's Norm Rating
Track1	0.1	0.2	0.7	0.58
Track2	0.9	0.1	0	0.58
Track3	0	0.1	0.9	?
Track4	0.2	0.7	0.1	0.58

<Topic-based CBF Example >

● Item-based CF Module

- ❖ 1. 유사도 계산
 - ◆ Normalized Rating으로부터 곡들간의 유사도 계산
 - ◆ 유사도 : Cosine 유사도
- ❖ 2. 이웃 선정
 - ◆ 유사도가 큰 상위 k 개의 음악 선택
- ❖ 3. Score 계산
 - ◆ 이웃의 Normalized Rating을 유사도로 가중 평균하는 Score 계산

$$\vec{r}_t = (r_{1t}, r_{2t}, \dots, r_{Mt})$$

$$score_{i,u,t} = \frac{\sum_{i=1}^k r_{u,t} \times sim(\vec{r}_t, \vec{r}_i)}{\sum_{i=1}^k |sim(\vec{r}_t, \vec{r}_i)|}$$

t : 추천대상이 되는 음악
 M : 전체 사용자 수
 k : 이웃 수
 i : 이웃 음악

	Track1	Track1	Track1	Track4
User1	0.58	0.58	?	0.58
User2	0.82	0.55	?	?
User3	0.03	?	1.00	0.07
User4	0.02	1.00	?	?

<Item-based CF Example>

● Hybrid Recommendation

- ❖ Topic-based CBF와 Item-based CF의 Score를 결합한 Hybrid Score 생성
- ❖ Hybrid Score가 높은 상위 n 개의 음악들을 사용자에게 추천

$$score_h = (1 - c) \times score_i + c \times score_t$$

c : Hybrid parameter

	Track1	Track1	Track1	Track4
User1			0.58	
User2			0.82	
User3		0.05		
User4			0.02	0.03

(a) Topic-based CBF

	Track1	Track1	Track1	Track4
User1			0.58	
User2			0.32	
User3		0.05		
User4			0.02	0.43

(b) Item-based CF



	Track1	Track1	Track1	Track4
User1			0.58	
User2			0.37	
User3		0.05		
User4			0.02	0.39

(c) Hybrid Recommendation

<Hybrid Score Example ($c = 0.1$)>

● 데이터 준비

❖ 사용자 청취 데이터

- ◆ 연구용으로 제공되는 Last.fm에서 수집된 사용자의 청취 이력 데이터 [25]
- ◆ 수집 기간 : 2005-05-02 ~ 2009-05-05
- ◆ 전체 사용자 수 : 992명
- ◆ 전체 Artist 수 : 107,397
- ◆ 전체 곡 수 : 961,416

< 사용자 청취 이력 Example (ex. user_000122) >

user_id	time	artist_id	artist	track_id	track
user_000122	2006-05-02 4:34	ea321799-9b1d-4e74-a074-a5facf597d82	Fugees	26e6f0c2-33b2-4024-a233-9e0dead277c0	Killing Me Softly
user_000122	2006-05-02 4:40	e2c00c56-8365-4160-9f40-a64682917633	The Goo Goo Dolls	0d390f24-9b0d-42d4-bbf7-3c4f7aeacb3b	Name
user_000122	2006-05-02 11:13	516cef4d-0718-4007-9939-f9b38af3f784	Fall Out Boy	786e1009-2663-49e0-873d-f0cf06f51a25	Dance, Dance
user_000122	2006-05-02 11:17	8d5c558e-44a8-4ab1-a4f5-98ae971f1029	Jump	39d46566-f733-4f8a-8d61-43d1d5e659b5	Body Parts
user_000122	2006-05-02 14:05	b1fb6a18-1626-4011-80fb-eaf83dfebc6	Jill Scott	1698bf78-fdaf-402e-aabd-42bcc4fd8564	Be Ready
user_000122	2006-05-02 14:09	d5be5333-4171-427e-8e12-732087c6b78e	The Black Eyed Peas	4886b9ac-493d-4817-9ded-9bfb6270310	Feel It
user_000122	2006-05-02 14:13	8d5c558e-44a8-4ab1-a4f5-98ae971f1029	Jump	4549cb65-1b8f-497f-9c06-46c7cbbc3b70	Rains In Asia
user_000122	2006-05-02 14:17	64b94289-9474-4d43-8c93-918ccc1920d1	Billy Joel	3e192415-3626-424d-b089-c39af522e08f	Say Goodbye To Hollywood
user_000122	2006-05-02 14:21	a3cb23fc-acd3-4ce0-8f36-1e5aa6a18432	U2	6b9a509f-6907-4a6e-9345-2f12da09ba4b	With Or Without You
user_000122	2006-05-02 14:25	e21857d5-3256-4547-afb3-4b6ded592596	Gorillaz	a5644f2a-b021-4c7a-85aa-1a1696333475	Every Planet We Reach Is Dead

● 데이터 준비

❖ 기간에 따른 데이터 분할

- ◆ Training Data : 사용자의 첫 1주일 청취 이력
- ◆ Test Data : 사용자의 첫 1주일 뒤 하루 동안의 청취 이력

❖ 사용자 Sampling

- ◆ 청취한 곡 중 id가 존재 하지 않은 곡이 10%미만인 사용자 제외
 - id가 존재하지 않는 곡들은 대부분 아시아 국가의 곡들로 Tag가 추출되지 않음
- ◆ 하루 동안 320곡 이상 청취한 사용자 제외
 - 하루 종일 음악을 청취하는 사용자는 이상치로 판단
 - 가정 : 하루 동안의 평균 수면 시간(8시간), 곡의 평균 재생 시간(3분)
 - $(24 - 8) \times 60 \div 3 = 320$ 곡

❖ Tag 수집

- ◆ 모든 사용자가 청취한 모든 곡들에 대하여 Last.fm API를 통해 Tag를 수집

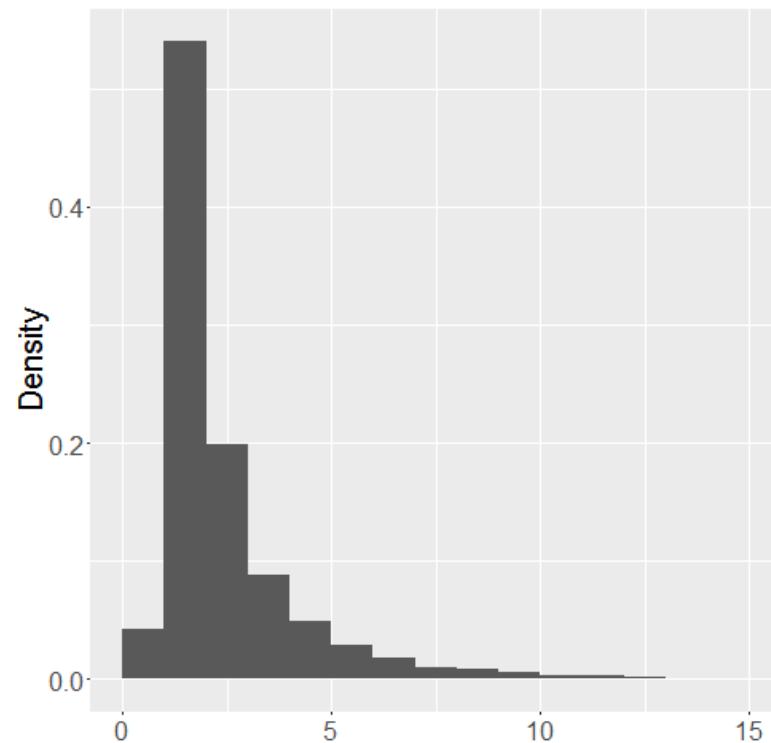
● 데이터 준비

❖ 곡 Sampling

- ◆ 청취한 사용자 수가 1명인 곡 제외
- ◆ Tag가 수집되지 않은 곡 제외
- ◆ Test Set에서 Training Set에 존재하지 않는 곡 제외

< 최종 Data >

	Training Data	Test Data
기간	1주일	1일
사용자 수	287	287
아티스트 수	1,848	838
곡 수	13,272	4,489
사용자의 하루 평균 청취 곡 수	20	21
사용자의 하루 평균 청취 횟수	32	25
곡당 평균 청취 사용자 수	3	1
곡의 평균 청취 횟수	5	2



< 곡당 청취 사용자 수 >

● 실험 방법

❖ LDA 파라미터

- ◆ Topic 수 : 30, 반복 수 : 2,000 ($\alpha : 0.1, \beta : 0.01$)

❖ Item-based CF & Topic-based CBF 파라미터

- ◆ 이웃 수(k) : 10, 20, 30, 40, 50 ($n = 10$)
- ◆ 추천 아이템(n) : 10, 20, 30, 40, 50

❖ hybrid Recommendation 파라미터

- ◆ $c : 0.1$

● 평가 척도

- ❖ 시스템의 성능을 측정하기 위해 Precision, Recall, F-measure를 척도로 선정
 - ◆ 사용자의 Item 선호 여부에 대한 척도로, 실제 산업에서 널리 적용되어짐 [26]

❖ Precision (P)

❖ Recall (R)

❖ F-measure (F)

	Irrelevant	Relevant
Not retrieved	TN	FP
Retrieved	FN	TP

$$P = \frac{TP}{FP + TP} \quad R = \frac{TP}{FN + TP} \quad F = \frac{2PR}{P + R}$$

< Precision, Recall, F-measure >

● Topic Modeling 결과

❖ Topic 종류

- ◆ 장르 관련 : 29개
 - Topic 1~13, Topic 14~30
- ◆ 분위기 관련 : 7개
 - Topic 2, 4, 7, 8, 14, 19, 29
- ◆ 국가 관련 : 3개
 - Topic 6, 12, 18
- ◆ 년도 관련 : 4개
 - Topic 12, 21, 24, 30
- ◆ 아티스트 특징 관련 : 1개
 - Topic 13

< Topic별 상위 5개 Tag 및 특징 >

Topic	Type	Tag
Topic 1	장르	metal, hard rock, heavi metal, thrash metal, nu metal
Topic 2	장르, 분위기	indi pop, indi, happi, swedish, fun
Topic 3	장르	hip hop, rap, rnb, christian, underground hip hop
Topic 4	장르, 분위기	trip hop, chillout, cover, downtempo, soundtrack
Topic 5	장르	funk, soul, jazz, funki, funk rock
Topic 6	장르, 국가	canadian, stoner rock, canadiangdchil, sludg, post metal
Topic 7	장르, 분위기	shoegaz, dream pop, ether, altern, ambient
Topic 8	장르, 분위기	electron, electro, danc, electroclash, electronica
Topic 9	장르	alt countri, folk, folk rock, countri, americana
Topic 10	장르	industri, industri metal, ebm, industri rock, electron
Topic 11	장르	metal, gothic metal, melod death metal, death metal, symphon metal
Topic 12	장르, 국가, 년도	indi, indi rock, lo fi, american i like, 1990
Topic 13	장르, 아티스트	femal vocalist, altern, singer songwrit, femal, femal vocal
Topic 14	분위기	beauti, mellow, sad, melancholi, love
Topic 15	장르	indi, indi rock, altern, rock, altern rock
Topic 16	장르	rock, garag rock, blue, blue rock, rock and roll
Topic 17	장르	electron, electronica, danc, techno, hous
Topic 18	장르, 국가	british, britpop, indi, rock, british i like
Topic 19	장르, 분위기	indi, mellow, indi pop, indi rock, altern
Topic 20	장르	instrument, experiment, post rock, ambient, iceland
Topic 21	장르, 년도	pop, danc, pop rock, rock, 1990
Topic 22	장르	post punk, experiment, nois rock, nois, sonic youth
Topic 23	장르	psychedel, psychedel rock, space rock, neo psychedelia, experiment
Topic 24	장르, 년도	1980, new wave, post punk, altern, rock
Topic 25	장르	rock, altern rock, altern, 1990, grung
Topic 26	장르	emo, rock, altern, pop punk, post hardcor
Topic 27	장르	progress rock, progress metal, rock, progress, radiohead
Topic 28	장르	punk, punk rock, ska, regga, rock
Topic 29	장르, 분위기	singer songwrit, folk, acoust, indi, mellow
Topic 30	장르, 년도	classic rock, rock, 1970, 1960, british

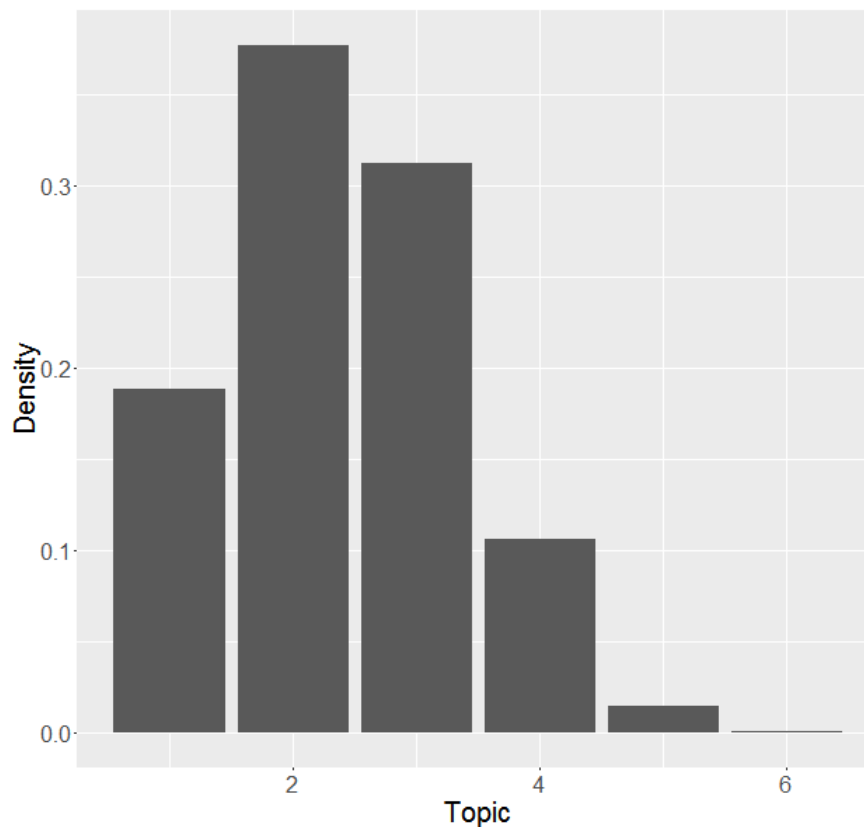
● Topic Modeling 결과

< 곡의 Topic 분포 Example >

Artist	Track	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8	Topic 9	Topic 10	Topic 11	Topic 12	Topic 13	Topic 14	Topic 15
Bob Dylan	If You See Her, Say Hello	0	0	0	0	0	0	0.01	0	0.37	0	0	0	0	0.15	0
Daft Punk	Something About Us	0	0	0	0.28	0.05	0	0	0	0	0	0	0	0	0.16	0
Damien Rice	The Blower'S Daughter	0	0	0	0.01	0	0	0	0	0	0	0	0	0	0.44	0.04
Green Day	American Idiot	0.02	0	0	0	0	0	0	0	0	0	0	0	0	0	0.03
Guns N' Roses	Sweet Child O' Mine	0.50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Jamiroquai	Virtual Insanity	0	0	0	0.02	0.57	0	0	0	0	0	0	0	0	0	0
Jason Mraz	Who Needs Shelter	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
John Mayer	Your Body Is A Wonderland	0	0	0	0	0	0	0	0	0	0	0	0	0	0.05	0
Michael Jackson	Man In The Mirror	0	0	0	0	0.14	0	0	0	0	0	0	0	0	0.03	0
Rihanna	Umbrella	0	0	0.46	0	0.01	0	0	0.01	0	0	0	0	0.12	0	0
		Topic 16	Topic 17	Topic 18	Topic 19	Topic 20	Topic 21	Topic 22	Topic 23	Topic 24	Topic 25	Topic 26	Topic 27	Topic 28	Topic 29	Topic 30
		0	0	0	0	0	0	0	0	0	0	0	0	0	0.31	0.15
		0	0.49	0	0	0	0.01	0	0	0	0	0	0	0	0	0
		0	0	0.01	0	0	0.05	0	0	0	0	0	0	0	0.46	0
		0	0	0	0	0	0.02	0	0	0	0.25	0.14	0	0.54	0	0
		0	0	0	0	0	0	0	0	0.14	0.06	0	0	0	0	0.30
		0	0.11	0.03	0	0	0.20	0	0	0	0.07	0	0	0	0	0
		0	0	0	0.18	0	0.14	0	0	0	0.03	0	0	0	0.65	0
		0	0	0	0	0	0.34	0	0	0	0.03	0	0	0	0.57	0
		0	0	0	0	0	0.65	0	0	0.17	0	0	0	0	0	0
		0	0	0	0	0	0.38	0	0	0	0	0	0	0	0	0

● Topic Modeling 결과

- ❖ 확률 값이 0.1 이상에 해당되는 Topic을 곡의 Topic으로 간주
 - ◆ 30개의 모든 Topic의 분포가 동일할 경우, 평균 값은 약 0.03
 - ◆ 곡의 평균 Topic수 : 2개

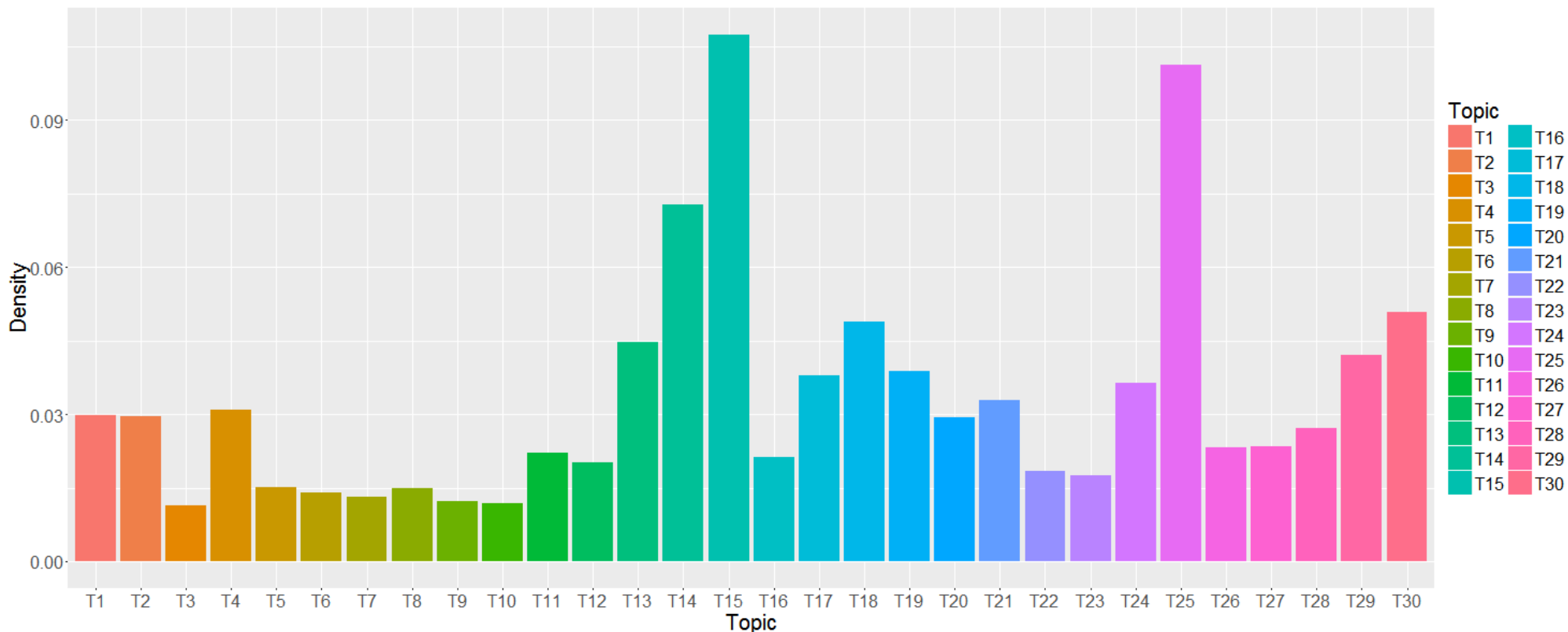


<곡당 Topic 수 >

● Topic Modeling 결과

❖ 전체 곡의 Topic 분포

- ◆ Topic 15과 Topic 250이 우세
 - 사용자들이 청취한 곡의 20%는 Indie rock과 Alternative rock
- ◆ Topic 14 : “슬픈” 분위기 관련 Topic으로 장르 관련 Topic과 더불어 자주 발생됨



● Topic Modeling 결과

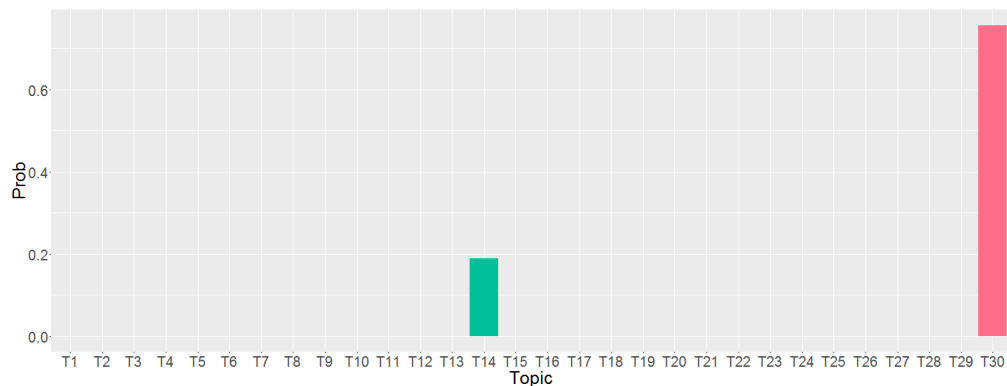
❖ Example

◆ 1) John Lennon – Imagine

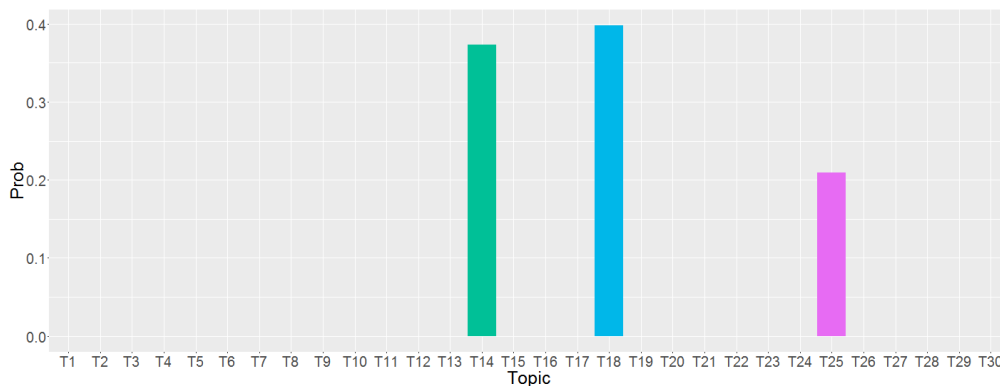
- Topic 14 : 슬픈 사랑 노래 (beautiful, mellow, sad, melancholy, love)
- Topic 30 : 60~70년대 rock (classic rock, rock, 1970, 1960, british)

◆ 2) Coldplay – The Scientist

- Topic 14 : 슬픈 사랑 노래 (beautiful, mellow, sad, melancholy, love)
- Topic 18 : 영국 노래 (british, britpop, indie, , rock, british i like)
- Topic 25 : 90년대 Alternative rock (rock, alternative rock, alternative, 1990, grunge)



(a) John lennon - Imagine



(b) Coldplay – The scientist

<곡의 Topic 분포 Example>

실험 및 성능 평가

● 실제 User 추천 결과

- ❖ Hybrid 추천은 Item-based CF와 Topic-based CBF를 동시에 고려하여 추천
 - ◆ Item-based CF : 인기 있는 곡 추천
 - ◆ Topic-based CBF : Topic이 유사한 곡 추천

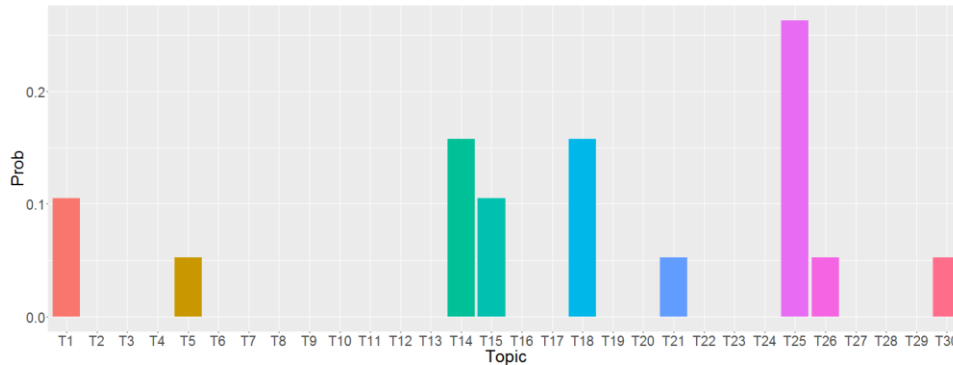
<사용자의 추천 리스트 (ex. user_000122)>

Rank	Training Set		Test Set		Score	Item_CF		Score	Topic_CBF		Score	Hybrid	
	Artist	Track	Artist	Track		Artist	Track		Artist	Track		Artist	Track
1	The All-American Rejects	Move Along	Chevelle	Closure	1	Linkin Park	What I've Done	1	Jimmy Eat World	Nothingwrong	0.925	Linkin Park	What I've Done
2	Coldplay	Trouble	Coldplay	Square One	0.662	Coldplay	Square One	1	My Chemical Romance	The Sharpest Lives	0.596	Coldplay	Amsterdam
3	Coldplay	We Never Change	Coldplay	Til Kingdom Come	0.5	Portishead	Roads	1	My Chemical Romance	Famous Last Words	0.538	Coldplay	The Scientist
4	Gorillaz	Dare	Elton John	Rocket Man	0.5	Ben Folds Five	Battle Of Who Could Care Less	1	My Chemical Romance	The End.	0.538	Coldplay	Talk
5	Peter Gabriel	The Book Of Love	Frank Sinatra	I've Got The World On A String	0.5	The Offspring	Staring At The Sun	1	The All-American Rejects	Dirty Little Secret	0.525	Coldplay	Square One
6	Gorillaz	All Alone	Jimmy Eat World	Sweetness	0.5	The Hives	Antidote	1	The Red Jump suit Apparatus	Home Improvement	0.508	Coldplay	Clocks
7	Pearl Jam	Better Man	Linkin Park	Numb	0.5	Linkin Park	Lying From You	1	The All-American Rejects	11:11 P.M.	0.508	Coldplay	Politik
8	Coldplay	X&Y	Snow Patrol	Run	0.5	Cat Power	Maybe Not	1	My Chemical Romance	House Of Wolves	0.500	Coldplay	Green Eyes
9	Coldplay	Twisted Logic			0.5	The Hives	B Is For Brutus	1	The All-American Rejects	Change Your Mind	0.5	Coldplay	Daylight
10	Red Hot Chili Peppers	Under The Bridge			0.5	Coldplay	The Hardest Part	1	Jimmy Eat World	Work	0.5	Trapt	Echo

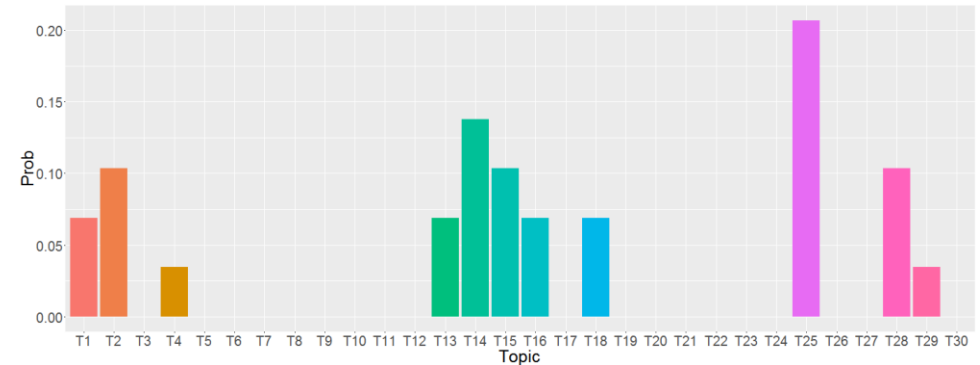
● 실제 User 추천 결과

❖ 추천 리스트의 전체 곡들의 각 Topic을 확률 분포로 표현

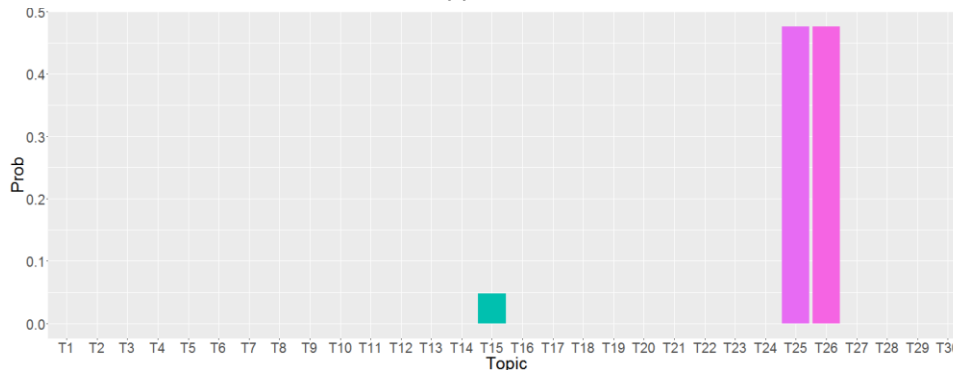
- ◆ Item-based CF : 다양한 Topic 존재
- ◆ Topic-based CBF : 특정 소수의 Topic 존재
- ◆ Hybrid : 두 알고리즘이 hybrid됨으로서 Item-based CF보다는 적은, Topic-based 보다는 많은 Topic이 존재



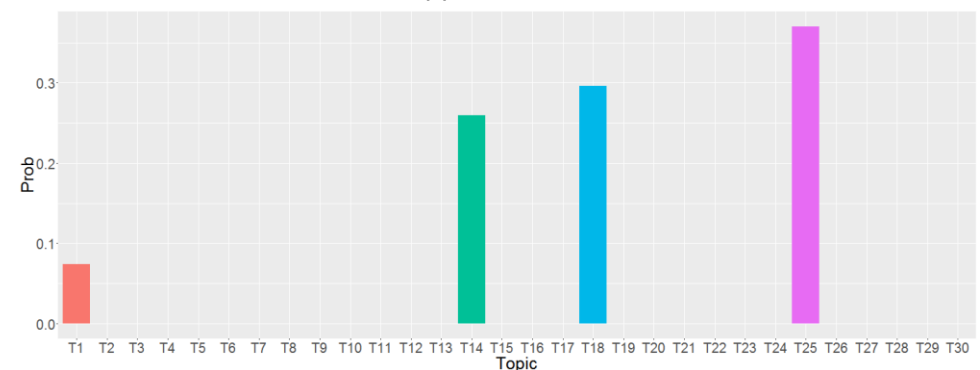
(a) Test Set



(b) Item-based CF



(c) Topic-based CBF



(d) Hybrid

● 성능 평가

❖ Paired t-test

- ◆ 각 Topic에 대하여, 추천리스트 곡들의 Topic 확률 값과 Test set의 Topic 확률 값의 차이를 검정

- H_0 : Topic 확률 값의 차이가 없다
- H_1 : Topic 확률 값의 차이가 있다
- 유의수준(α) : 0.05

◆ 결과

- Item-based CF : Topic 17, 25에서 Topic값의 차이가 존재한다고 볼 수 있음
- Topic-based CF : Topic 9에서 차이가 존재한다고 볼 수 있음
- Hybrid : 모든 Topic에 대하여 차이가 있다고 볼 수 없음

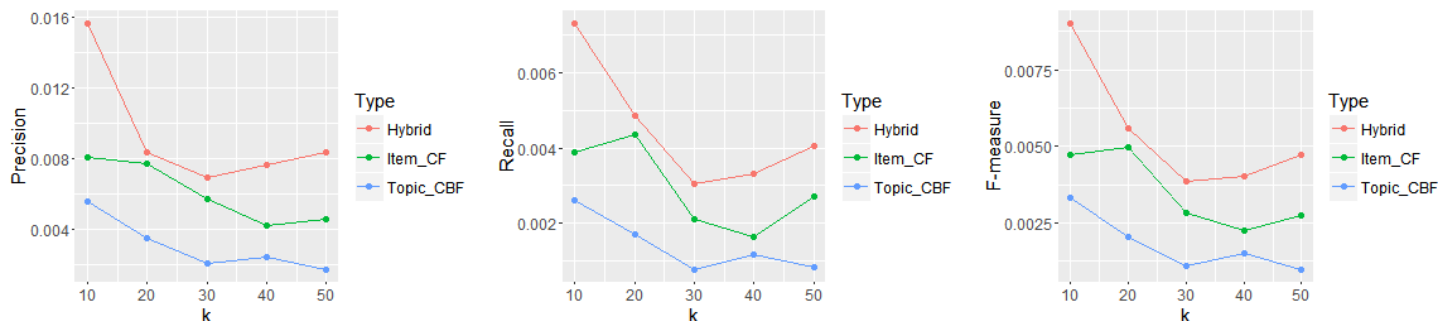
< Topic별 Paired t-test 결과 >

Topic	Item_CF		Topic_CBF		Hybrid	
	t value	p value	t value	p value	t value	p value
Topic 1	0.04	0.97	1.26	0.21	0.90	0.37
Topic 2	1.59	0.11	0.59	0.56	1.30	0.19
Topic 3	0.42	0.67	0.76	0.45	1.14	0.26
Topic 4	-1.22	0.22	-0.97	0.33	-1.27	0.21
Topic 5	-0.25	0.81	0.62	0.53	0.57	0.57
Topic 6	-0.73	0.46	0.40	0.69	-1.18	0.24
Topic 7	0.79	0.43	1.15	0.25	0.68	0.50
Topic 8	0.06	0.95	-1.61	0.11	-0.29	0.77
Topic 9	-1.28	0.20	-2.48	0.01	-0.29	0.77
Topic 10	-1.06	0.29	1.56	0.12	-0.29	0.78
Topic 11	-0.86	0.39	-1.55	0.12	-1.80	0.07
Topic 12	-0.15	0.88	-0.70	0.48	-0.64	0.52
Topic 13	-0.11	0.92	1.24	0.22	-0.35	0.73
Topic 14	-0.59	0.56	-1.02	0.31	-1.19	0.23
Topic 15	0.39	0.69	1.23	0.22	-0.42	0.67
Topic 16	-0.32	0.75	0.07	0.95	0.51	0.61
Topic 17	2.36	0.02	0.83	0.40	1.18	0.24
Topic 18	0.50	0.62	1.34	0.18	-0.74	0.46
Topic 19	1.69	0.09	-0.27	0.79	2.00	0.05
Topic 20	1.15	0.25	-0.58	0.56	-0.03	0.98
Topic 21	-0.64	0.52	-1.22	0.23	-0.47	0.64
Topic 22	-0.08	0.93	-0.15	0.88	-0.49	0.62
Topic 23	0.13	0.89	-0.56	0.58	-0.03	0.98
Topic 24	1.41	0.16	-0.23	0.82	0.88	0.38
Topic 25	-2.78	0.01	-0.16	0.87	-1.35	0.18
Topic 26	-0.77	0.44	-0.66	0.51	-0.22	0.83
Topic 27	0.38	0.71	0.70	0.48	0.20	0.84
Topic 28	-0.14	0.89	1.10	0.27	-0.37	0.71
Topic 29	0.98	0.33	-1.22	0.22	1.59	0.11
Topic 30	0.06	0.95	-0.54	0.59	1.75	0.08

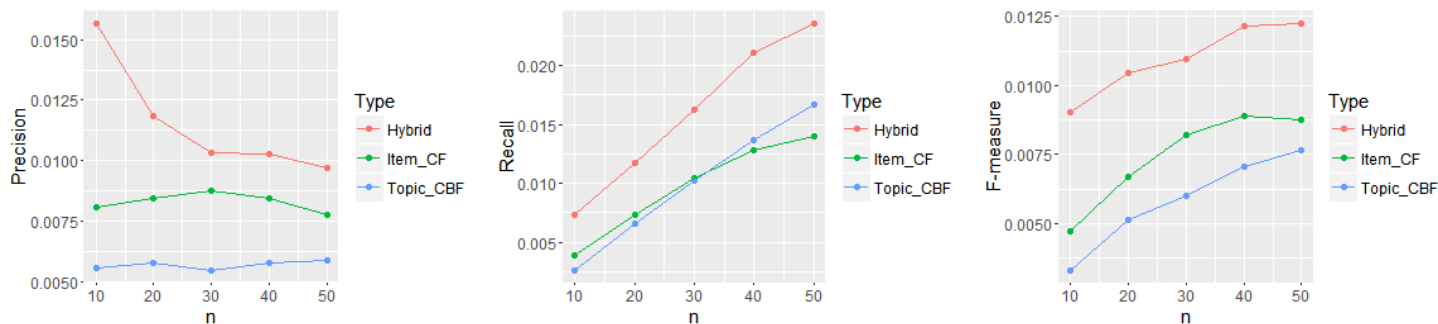
● 성능 평가 (Cont.)

❖ Precision, Recall, F-measure

- ◆ Topic-based CBF은 Item-based CF 보다 성능이 좋지 않으나, 추천 아이템 수 가 30개 이상일 경우 Recall이 더 우수함
- ◆ Hybrid 추천 방법이 모든 척도에서 성능이 좋음
 - 이웃수가 10 ($k = 10$), 추천 아이템 수가 10 ($n = 10$) 일 때 가장 좋은 성능을 보임



(a) 이웃 수 증가에 따른 Precision, Recall, F-measure



(a) 추천 아이템 수 증가에 따른 Precision, Recall, F-measure

< 알고리즘별 Precision, Recall, F-measure >

● 향후 연구

- ❖ 연구용 데이터셋의 과거 사용자가 아닌 현재 사용자에게 대한 적용 필요
 - ◆ 본 연구에서는 사용자들이 곡을 청취한 시기와 수집된 Tag의 시기가 일치하지 않음
- ❖ CBF에서 Item의 속성으로 다른 정보와의 같이 이용
 - ◆ 기본적인 곡의 Description, 가사 등과 같은 다른 정보와 Topic을 동시에 적용
- ❖ Tag에 대한 체계적인 분류 과정이 필요
 - 장르와 같이 얻기 쉬운 정보가 아닌 사용자의 의견과 같은 Tag에만 존재하는 정보를 중점적으로 이용을 할 필요가 있음
- ❖ 다른 Hybrid 방법에 대한 추가적인 연구 필요
 - ◆ Weighted 방식이 아닌 다른 방식의 알고리즘 결합 연구

- [1] Dror, G., Koenigstein, N., Koren, Y., Weimer (2012), M.: The Yahoo! Music Dataset and KDDCup' 11. Journal of Machine Learning Research: Proceedings of KDD-Cup 2011 competition 18, 3-18
- [2] Levy, M., & Sandler, M. (2008). Learning latent semantic models for music from social tags. Journal of New Music Research, 37(2), 137-150.
- [3] Levy, M., & Sandler, M. (2007). A semantic space for music derived from social tags. Austrian Computer Society, 1, 12.
- [4] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13(5-6), 393-408.
- [5] Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999, August). An algorithmic framework for performing collaborative filtering. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (pp. 230-237).
- [6] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295).
- [7] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User modeling and user-adapted interaction, 12(4), 331-370.
- [8] Krulwich, B. (1997). Lifestyle finder: Intelligent user profiling using large-scale demographic data. AI magazine, 18(2), 37
- [9] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. (1999, August). Combining content-based and collaborative filters in an online newspaper. In Proceedings of ACM SIGIR workshop on recommender systems(Vol. 60).
- [10] Shardanand, U., & Maes, P. (1995, May). Social information filtering: algorithms for automating "word of mouth" . In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 210-217).
- [11] Iwahama, K., Hijikata, Y., & Nishida, S. (2004, January). Content-based filtering system for music data. In Applications and the Internet Workshops, 2004. SAINT 2004 Workshops, 2004 International Symposium on (pp. 480-487).
- [12] Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In Advances in Neural Information Processing Systems (pp. 2643-2651).
- [13] 김현희, 김동건, & 조진남. (2013). 사용자 청취 습관과 태그 정보를 이용한 하이브리드 음악 추천 시스템. 한국컴퓨터정보학회논문지, 18(2), 107-116.
- [14] Hariri, N., Mobasher, B., & Burke, R. (2012, September). Context-aware music recommendation based on latent topic sequential patterns. In Proceedings of the sixth ACM conference on Recommender systems (pp. 131-138).
- [15] Zhang, Y. C., Séaghdha, D. Ó., Quercia, D., & Jambor, T. (2012, February). Auralist: introducing serendipity into music recommendation. In Proceedings of the fifth ACM international conference on Web search and data mining (pp. 13-22)
- [16] Blei, D. M., & Lafferty, J. D. (2007). A correlated topic model of science. *The Annals of Applied Statistics*, 17-35.

- [17] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6), 391.
- [18] Hofmann, T. (1999, August). Probabilistic latent semantic indexing. *In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 50–57).
- [19] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan), 993–1022.
- [20] Bischoff, K., Firan, C. S., Nejdl, W., & Paiu, R. (2008, October). Can all tags be used for search?. *In Proceedings of the 17th ACM conference on Information and knowledge management* (pp. 193–202).
- [21] Geleijnse, G., Schedl, M., & Knees, P. (2007, September). The Quest for Ground Truth in Musical Artist Tagging in the Social Web Era. *In ISMIR* (pp. 525–530). Hu, X., Downie, J. S., & Ehmann, A. F. (2009)
- [22] Hu, X., & Downie, J. S. (2010, June). Improving mood classification in music digital libraries by combining lyrics and audio. *In Proceedings of the 10th annual joint conference on Digital libraries* (pp. 159–168).
- [23] Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130–137.
- [24] Jawaheer, G., Weller, P., & Kostkova, P. (2014). Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(2), 8.
- [25] Celma, O., *Music Recommendation and Discovery in the Long Tail*, Springer (2010)
- [26] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1), 5–53.