

# 음악과 플레이리스트의 메타데이터를 활용한 하이브리드 음악 추천 시스템에 관한 연구

이현태

한양대학교 비즈니스 인포매틱스학과  
(leeh0113@hanyang.ac.kr)

임규건

한양대학교 경영대학  
(gglim@hanyang.ac.kr)

추천 시스템은 인터넷의 발달로 급격하게 증가하는 정보의 양으로 인해 생긴 정보 선택의 어려움을 소비자에게 덜어주고 각 개인의 취향에 맞는 정보를 효율적으로 보여주는 중요한 역할을 한다. 특히, E-commerce와 OTT 기업은 상품과 콘텐츠 양이 급격하게 증가하면서 추천 시스템의 도움 없이는 인기 있는 상품만 소비되는 현상을 극복하지 못한다. 이러한 현상을 극복하고 고객 개인 취향에 맞는 정보 혹은 콘텐츠를 제공해 고객의 소비를 유도하기 위해 추천 시스템의 연구가 활발히 진행되고 있다.

일반적으로 유저(user)의 과거 행동 이력을 활용한 협업 필터링이 유저가 선호한 콘텐츠의 정보를 활용하는 콘텐츠 기반 필터링에 비해 높은 성능을 보여준다. 하지만 협업 필터링은 과거 행동 데이터가 부족한 유저에 대해서는 추천의 성능이 낮아지는 콜드 스타트(Cold Start) 문제를 겪게 된다.

본 논문에서는 카카오 아레나 경진대회에서 주어진 음악 스트리밍 서비스 멜론의 플레이리스트 데이터를 기반으로 앞에서 언급한 콜드 스타트 문제를 해결할 수 있는 하이브리드 음악 추천 시스템을 제시했다. 본 연구에서는 플레이리스트에 수록된 곡 목록과 각 음악과 플레이리스트의 메타데이터를 활용해 절반 혹은 전부 가려진 플레이리스트의 다른 수록 곡을 예측하는 것을 목표로 하였다. 이를 위해 플레이리스트 안에 곡이 있는 경우와 아예 곡이 없는 경우를 나눠서 추천을 진행하였다.

플레이리스트 안에 곡이 있는 경우에는 해당 플레이리스트의 곡 목록과 각 곡의 메타데이터를 활용하기 위해 LightFM을 활용하였다. 그 다음에 Item2Vec을 활용해 플레이리스트에 있는 수록 곡과 태그 및 제목의 임베딩 벡터를 생성하고 이를 추천에 활용하였다. 최종적으로 LightFM과 Item2Vec 모델의 앙상블을 통해 최종 추천 결과를 생성하였다.

플레이리스트 안에 곡이 없고 태그 혹은 제목만이 존재할 경우에는 플레이리스트의 메타데이터인 태그와 제목을 FastText를 활용해 사전 학습을 시켜 생성된 플레이리스트 벡터를 기반으로 플레이리스트 간의 유사도를 활용하여 추천을 진행하였다.

이렇게 추천한 결과, 기존 Matrix Factorization(MF)에서 해결하지 못한 콜드 스타트 문제를 해결할 수 있었을 뿐만 아니라 곡과 플레이리스트의 메타데이터를 활용해 기존 MF 모델인 ALS와 BPR 그리고 Word2Vec 기반으로 추천해 주는 Item2Vec 기술보다 높은 추천 성능을 낼 수 있었다. 또한, LightFM을 토대로 다양한 곡의 메타데이터를 실험한 결과, 여러 메타데이터 중에서 아티스트 정보를 단독으로 활용한 LightFM 모델이 다른 메타데이터를 활용한 LightFM 모델들과 비교해 가장 높은 성능을 보여준다는 것을 확인할 수 있었다.

**주제어** : 하이브리드 추천 시스템, 콜드 스타트, LightFM, Item2Vec, FastText, 사전학습

논문접수일 : 2023년 5월 7일

논문수정일 : 2023년 7월 28일

게재확정일 : 2023년 8월 25일

원고유형 : Regular Track

교신저자 : 임규건

## 1. 서론

최근 빅데이터 시대의 도래로 하루에 접할 수 있는 정보량은 기하급수적으로 증가하고 있다.

하지만 이러한 수많은 정보 속에서 각 개인의 상황과 취향에 맞는 정보를 찾는다는 것은 더욱 어려워졌고, 이 모든 정보를 개인이 모두 찾아보는 것은 불가능한 일에 가까워졌다. 따라서 많은

기업들이 추천 시스템을 활용해 각 소비자의 취향에 맞는 정보를 제공함으로써 정보 검색에 대한 피로도를 줄임과 동시에 소비를 유도하는 전략에 관심을 가지게 되었다. 그 중에서도 Amazon과 Netflix는 추천 시스템을 적용해 큰 성공을 이룬 기업의 대표적인 사례로 손꼽힌다.

E-commerce, OTT 기업뿐만 아니라 최근에는 음악 스트리밍 기업에서도 추천 시스템의 중요성이 커지고 있다. 이러한 추세에 맞춰서 Spotify는 유저(user) 개인의 취향에 맞는 음악의 목록을 제시해주는 플레이리스트(playlist) 추천으로 유명한 대표적인 기업이다. 하지만 음악 플레이리스트 추천에서도 다른 도메인(domain)과 마찬가지로 아이템(item) 소비 이력이 아예 없는 새로운 유저 혹은 소비 이력이 적은 유저에 대해서는 추천의 성능이 떨어지는 콜드 스타트(Cold Start) 문제를 피할 수가 없다. 또한, 음악 도메인에서는 Netflix와 같은 OTT 플랫폼과는 다르게 유저가 특정 음악에 평점을 내리기 보다는 재생 이력만 남기는 경우가 많기 때문에 기존 추천 시스템에서 흔히 평점을 예측하는 문제보다는 사용자가 소비할 가능성이 높은 아이템들을 순서대로 나열하는 문제(Top-N Recommendation)라고 할 수 있다. 이러한 배경에 발맞춰서 카카오에서는 2020년에 멜론의 음악과 플레이리스트의 데이터를 활용해 곡의 일부 혹은 전부가 가려졌을 때와 같은 다양한 상황에서 특정 플레이리스트의 수록 곡을 예측하는 플레이리스트 추천 경진대회를 열었다. 해당 데이터는 카카오 아레나 홈페이지에 공개되어 이용약관 동의 하에 다운로드 받을 수 있다(Ferrao et al., 2021).

본 논문에서는 앞서 언급한 경진대회에서 주어진 음악과 플레이리스트 데이터를 활용해 상황에 따라 추천 방법을 달리하는 하이브리드

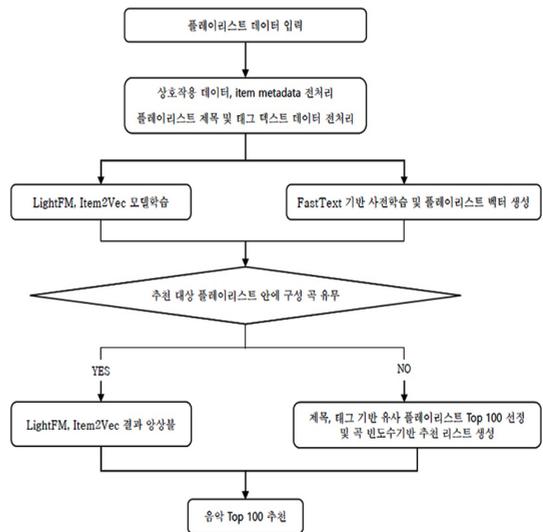
(Hybrid) 추천 전략을 제시하고자 한다. 즉, 플레이리스트에 수록 곡이 있는 경우에는 플레이리스트의 수록 곡 ID와 곡과 플레이리스트의 메타데이터를 활용한 LightFM과 Item2Vec의 앙상블 모델을 활용하고, 플레이리스트에 수록 곡이 없고 플레이리스트의 태그 혹은 제목만 있는 경우에는 유사한 태그와 제목을 가진 플레이리스트를 활용해 추천하는 방안을 제안하고자 한다. 마지막으로, 여러 음악의 메타데이터를 추천에 활용함으로써 어떠한 곡의 메타데이터가 추천 성능 향상에 가장 기여를 많이 하는지를 탐색하고자 한다.

최근, 사용자와 아이템 간의 상호작용 데이터뿐만 아니라 텍스트로 이루어진 아이템의 메타데이터를 추천에 활용하는 선행연구가 많이 진행됐다. 김동환(2023)은 OTT 플랫폼 콘텐츠의 내용적인 부분을 설명하는 제목과 시놉시스와 같은 텍스트로 이루어진 메타 데이터를 활용해 유사한 콘텐츠를 추천하기 위해 한국어 언어모델인 KLUE-RoBERTa-large를 활용하였다. 장동수 등(2023)은 딥러닝 기반 추천 시스템 모델에 온라인 리뷰를 활용하여 고객의 선호도를 정교하게 학습할 수 있는 추천 방법론을 제안하였다. 이를 위해 장동수 등(2023)은 고객과 상품 간 복잡한 상호작용을 비선형으로 학습할 수 있는 상호작용 네트워크를 구성하고 고객의 온라인 리뷰의 특징을 추출하는 언어학적 요소 네트워크를 구성했다. 마지막으로 상호작용 네트워크와 언어학적 네트워크가 출력된 벡터를 결합하여 연산을 수행하는 선호도 예측 네트워크를 활용하여 고객의 선호도 평점을 예측하였다. 홍태호 등(2023)은 온라인 리뷰의 상품 속성과 고객 속성을 통합한 지능형 추천 시스템을 제안하였다. 이를 위해 홍태호 등(2022)은 영화 속성을 추출하기 위해 LDA를 적용하고 고객 속성을 추출하기

위해 KoBERT를 적용하여 이들을 통합한 지능형 추천 시스템을 제안하였다. 본 논문에서는 플레이리스트의 제목과 태그를 학습해 플레이리스트 벡터를 생성하고 OOV(Out Of Vocabulary) 문제에 대응하기 위해 BERT 기반 모델을 활용한 김동환(2023)과 홍태호(2023)과 다르게 FastText 기법을 활용하였다. 이렇게 생성된 플레이리스트 벡터는 플레이리스트 안에 구성 곡이 없는 경우에 활용된다. 또한, LightFM과 Item2Vec을 앙상블하기 위해 활용된 플레이리스트 벡터는 플레이리스트 내의 수록 곡 ID와 토큰화된 플레이리스트의 제목과 태그의 텍스트 메타 데이터를 활용하여 Word2Vec 기법을 기반으로 플레이리스트 벡터를 생성하였다.

## 2. 연구 방법

본 논문에서는 콜드 스타트 문제에 대응하고 주어진 메타데이터를 활용하기 위하여 Switching (Burke et al., 2002) 전략을 바탕으로 추천 시스템을 구성하였다. 따라서 플레이리스트 안에 수록 곡이 있는 경우에는 LightFM(Kula et al., 2015)과 Item2Vec(Barkan et al., 2016) 모델의 결과를 앙상블한 추천 리스트를 제공하고, 수록 곡이 없고 태그 혹은 제목만 있을 경우, 텍스트 데이터를 FastText(Bojanowski et al., 2017) 기반으로 사전 학습을 시켜 플레이리스트 벡터를 만들고 유사한 플레이리스트 상위 100개를 선정해 그 안에 수록된 곡 빈도수 순서대로 추천을 진행한다. 해당 추천 시스템의 전체적인 흐름도는 <그림 1>과 같다.



<그림 1> 본 연구의 하이브리드 추천 시스템 Flow Chart

<그림 1>의 Switching 기법의 하이브리드 추천 시스템 전략을 설명하면, 먼저 플레이리스트의 제목과 태그와 같은 메타데이터 그리고 수록 곡이 포함된 데이터가 입력이 된다. 그 이후에 수록 곡 ID와 수록 곡의 메타데이터는 상호작용 데이터로 여겨 전처리를 진행하고 플레이리스트의 태그와 제목과 같은 텍스트 데이터는 불용어 제거와 토큰화와 같은 전처리 과정을 거친다. 이렇게 전처리 된 수록곡 ID와 수록 곡의 메타데이터를 활용하여 LightFM 모델을 학습하고, 수록 곡 ID와 제목 및 태그 데이터를 활용하여 Item2Vec 기법으로 플레이리스트 임베딩 벡터를 생성한다. 또한, 제목과 태그와 같은 텍스트 데이터를 활용하여 FastText 기법으로 플레이리스트 벡터를 생성한다. 추론 과정에서는, 플레이리스트 안에 구성 곡 유무에 따라 다르게 이루어진다. 구성 곡이 있는 경우에는 앞서 언급한 LightFM과 Item2Vec을 활용하여 추천 리스트를 생성하고, 구성 곡이 없는 경우에는 앞서 언급한 FastText 기법으로

생성한 플레이리스트 벡터를 코사인 유사도 기반으로 추천 리스트를 생성한다.

## 2.1 데이터 설명 및 구성

본 논문에서 활용한 데이터는 ‘카카오 아레나’ 경진대회에서 주어진 멜론 플레이리스트 데이터이다 (Ferraro et al., 2021). 해당 경진대회에서는 모델 훈련에 사용하게 될 train 데이터와 경진대회 제출용으로 주어진 test, validation 데이터가 주어졌다 (Ferraro et al., 2021). 또한, 해당 경진대회에서 플레이리스트에 수록된 모든 곡에 대한 메타데이터를 json 파일로 주어졌다. 본 논문에서는 성능 평가를 하기 위해서 플레이리스트 원본 데이터가 담긴 train 데이터 셋을 사용해 연구를 진행하고 곡에 대한 메타데이터 파일을 LightFM의 아이템 특성 데이터로 활용하였다. 본 논문에서 활용되는 플레이리스트 train 데이터에 대한 구성은 <표 1>과 같다.

<표 1> 플레이리스트 train 데이터 구성표

필드명	설명
tags	플레이리스트에 매핑된 태그 리스트
id	플레이리스트 id
plylst_title	플레이리스트 제목
songs	플레이리스트에 수록된 곡 id 리스트
like_cnt	좋아요 개수
updt_date	플레이리스트 수정 날짜

해당 경진대회에서 주어진 플레이리스트 메타데이터에서는 플레이리스트를 만든 사용자가 제출한 플레이리스트 제목과 태그 그리고 마지막으로 수정한 날짜와 다른 사용자로부터 받은 좋아요 개수가 포함되어 있다. 해당 경진대회에서는 총 707,989개의 곡에 대한 메타데이터를 공개

했다. 이 중에서 본 논문에서는 곡 장르 리스트, 곡 아티스트 명 리스트, 곡 발매일을 활용해 어떠한 메타데이터가 추천 성능 향상에 가장 기여를 많이 하는지 탐색하였다.

경진대회에서 제출용으로 주어진 test와 validation 데이터에서는 플레이리스트 안에 수록 곡이 없는 경우가 다수 존재한다(Ferraro et al., 2021). 따라서 본 논문에서도 비슷한 상황에서의 추천 시스템 성능을 실험하기 위해서 기존에 주어진 train 데이터를 분리하여 실험을 진행했다. 먼저 train 데이터에서 80%(훈련용 데이터)는 플레이리스트의 모든 데이터가 있도록 하고, 나머지 20%는 검증용 데이터로 플레이리스트 안에 있는 수록 곡을 절반 혹은 전부 가리도록 했다(검증용 데이터). 분리된 데이터의 자세한 구성은 <표 2>와 같다.

<표 2> 학습 및 평가를 위해 분리된 데이터 셋의 구성표

데이터	구성	비율	개수
훈련용 데이터	모든 데이터	0.8	92,056
검증용 데이터	곡 데이터 절반 마스킹	0.06	6,904
	곡과 태그 데이터 절반 마스킹	0.10	11,508
	태그 데이터 절반만 보존	0.03	3,452
	제목 데이터만 보존	0.01	1,151
정답 데이터	마스킹된 데이터	0.2	23,015

<표 2>에서 명시된 것처럼, 검증용 데이터에서는 전체 6% 비율로 수록 곡 절반을 마스킹하고 10% 비율로 곡과 태그 데이터 모두 절반 마스킹했다. 또한, 나머지는 태그 절반과 제목 데이터만 보존되도록 하고 이는 플레이리스트 안에 있는 수록 곡이 전부 마스킹이 된 데이터셋이다. 검증용 데이터 셋에서 마스킹이 된 플레이리스트의 수록

곡들은 정답 데이터 셋 (정답 데이터)에 저장하였다. 본 논문에서는 훈련용 그리고 검증용 데이터에 있는 수록 곡과 메타데이터를 모델 학습에 사용하고 이렇게 학습된 모델을 토대로 정답 데이터에 있는 수록 곡을 추정하는 것을 목적으로 하였다.

본 논문에서는 다양한 상황에서 추천이 가능한 하이브리드 추천 시스템을 구축하는 것을 목적으로 삼았다. 이를 위해, 플레이리스트의 메타 데이터인 제목과 태그 그리고 음악의 메타 데이터를 동시에 활용하는 방안을 모색한다. 따라서 훈련과 검증 세트의 구성을 달리해 학습을 진행하는 교차 검증을 수행하는 대신 <표 2>처럼 검증용 데이터의 구성을 달리하는 방법으로 학습을 진행했다. <표 2>처럼 검증용 데이터를 달리하면서 교차 검증을 실행하기에는 데이터 구성이 너무 복잡함이 따르므로 교차 검증을 실행하지 않은 것은 본 논문의 한계점으로 지적될 수 있는 부분이다.

## 2.2 데이터 전처리 및 모델 학습

### 2.2.1 ALS, BPR 학습 및 데이터 전처리

먼저 ALS(Koren et al., 2009), BPR(Rendle et al., 2009)와 같은 MF 기반 알고리즘을 베이스라인 모델을 구현하기 위해서 <그림 2>와 같이 사용자-아이템 상호작용 행렬을 구성하였다.

User/Item	Item 1	Item 2	Item 3	...	...	Item N-1	Item N
User 1	0	1	0	...	...	0	0
User 2	1	0	0	...	...	1	0
User 3	0	0	1	...	...	0	0
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
User M-1	1	0	1	...	...	0	1
User M	0	0	1	...	...	0	0

<그림 2> 사용자-아이템 상호작용 행렬(Matrix)

<그림 2>에서 user는 플레이리스트 ID에 해당하고 item은 곡 ID에 해당한다. 또한, 본 논문에서는 플레이리스트 안에 특정 곡의 구성유무를 표현하는 행렬이므로 암시적 피드백 데이터에 해당한다. 따라서 <그림 2>처럼 특정 플레이리스트 안에 특정 곡이 있을 경우에는 1로 표시하고 포함하지 않을 경우에는 해당 곡 ID에 해당하는 공간에 0으로 표시하였다.

본 논문에서 활용하는 데이터에는 총 707,989개의 곡 데이터가 존재하지만 각 플레이리스트마다 평균적으로 46개의 곡이 포함되어 있다. 따라서 이는 희소율이 매우 큰 희소 행렬이므로 <그림 2>와 같이 행렬을 그대로 생성한다면 메모리 소요가 매우 클 것이다. 이러한 문제에 대응하기 위해서 본 논문에서는 SciPy(Virtanen et al., 2020) 라이브러리를 활용해 압축 희소 열 행렬(Compressed Sparse Row(CSR) Matrix)로 플레이리스트-곡 상호작용 행렬을 만들어 ALS와 BPR 알고리즘의 학습 데이터로 활용하였다.

ALS와 BPR 학습을 위해 설정한 하이퍼파라미터는 <표 3>과 같다. 아래 표에 설정되지 않은 하이퍼파라미터는 Implicit(Frederickson, B., 2017) 라이브러리에 주어진 기본값으로 설정했다.

<표 3> ALS, BPR 학습을 위한 설정 파라미터

알고리즘 종류	하이퍼파라미터 명	설정 값
ALS	차원 수(factors)	100
	정규화(regularizations)	0.1
	훈련 횟수(Iterations)	200
BPR	차원 수(factors)	100
	정규화(regularization)	0.1
	훈련 횟수(Iterations)	2000

이 알고리즘으로 사용자 잠재요소와 아이템 잠재

요소의 내적 값으로 추천 대상 플레이리스트 ID 별로 각 아이터마다 점수를 구하였다. 그 다음, 추천 대상 플레이리스트 ID의 수록 곡을 제외하고 점수가 높은 상위 100개 곡을 추천 리스트로 제공하였다.

## 2.2.2 LightFM 학습 및 데이터 전처리

본 논문에서 곡 아이터의 메타데이터를 사용하기 위해 활용하게 될 추천 시스템 알고리즘인 LightFM은 유저와 아이터를 각각의 콘텐츠 특성의 잠재요소로 표현한 하이브리드 Matrix Factorization 모델이다(Kula, 2015). 여기서 유저와 아이터의 잠재 요소는 각 특성의 잠재 벡터의 합으로 정의되고 편향도 각 특성의 편향의 합으로 정의된다. 여기서 유저  $u$ 의 특성은  $f_u \subset F^U$ , 아이터  $i$ 의 특성은  $f_i \subset F^I$ , 유저의 잠재요소는  $q_u$ , 아이터의 잠재요소는  $p_i$ , 유저 특성의 임베딩 벡터는  $e_f^U$ (편향은  $b_f^U$ ), 아이터 특성의 임베딩 벡터는  $e_f^I$ (편향은  $b_f^I$ )로 표현하면 해당 식은 아래와 같이 정의된다.

$$q_u = \sum_{j \in f_u} e_j^u p_i = \sum_{j \in f_i} e_j^i$$

$$b_u = \sum_{j \in f_u} b_j^U b_i = \sum_{j \in f_i} b_j^I$$

LightFM 모델의 유저  $u$ 와 아이터  $i$ 의 평점 및 선호도 예측은 아래와 같이 유저와 아이터 벡터의 내적으로 이루어지고 유저와 아이터 특성의 편향에 의해 조정된다(Kula, 2015)

$$\hat{r}_{ui} = f(q_u \cdot p_u + b_u + b_i)$$

본 논문에서는 플레이리스트 안에 특정 곡이 포함된 여부를 파악하는 문제이므로 암시적 피드백 데이터이고 이는 이진 데이터에 해당한다.

따라서 Kula(2015)이 이진 데이터 예측을 위해 활성화 함수로 시그모이드 함수를 사용한 것과 똑같이 본 논문에서도 해당 함수를 선호도 예측 함수의 활성화 함수로 사용한다.

$$f(x) = \frac{1}{1 + \exp(-x)}$$

해당 알고리즘의 최적화는 데이터에 대한 아래와 같은 함수를 최대화하는 것으로 정의된다. 아래 수식에서  $(u, i) \in U \times I$ 는 유저와 아이터 상호작용 쌍으로 정의되고 긍정적 상호작용( $S^+$ )과 부정적 상호작용( $S^-$ )의 합집합이다(Kula, 2016).

$$L(e^u, e^I, b^U, b^I) = \prod_{(u, i) \in S^+} \hat{r}_{ui} \times \prod_{(u, i) \in S^-} (1 - \hat{r}_{ui})$$

이렇게 구성된 LightFM 알고리즘으로 유저와 아이터의 메타데이터를 활용함으로써 콜드 스타트나 상호작용 데이터의 희소성에 대응할 수 있다(Kula, 2015).

따라서, LightFM에서는 ALS와 BPR과는 다르게 상호작용 행렬을 만들고 추가적으로 아이터과 유저에 대한 특성 행렬을 생성하였다. 아이터의 특성 행렬 역시 크기가 크므로 CSR 형태로 생성해 학습에 사용한다. 이 때 아이터 특성 행렬은 (총 아이터 개수) (총 아이터 개수 + 특성 개수)의 크기를 가진다. 본 논문에서는 곡의 메타데이터의 구성을 달리해 어떠한 메타데이터의 구성이 LightFM 기준으로 기존 MF(ALS, BPR)보다 높은 성능을 내고 추천 성능 향상에 가장 기여를 많이 하는지 탐색하였다. 따라서 각 메타데이터의 구성에 따라 아이터 특성 행렬의 크기를 달리해 학습을 진행하여 성능을 비교 분석하였다. 각 메타데이터의 구성에 따른 아이터 특성 행렬과 상호작용 행렬의 크기는 <표 4>와 같다.

〈표 4〉 LightFM 모델에 활용할 상호작용과 특성 행렬의 크기 및 원소 개수

행렬 종류	행렬 크기	원소 개수
플레이리스트-곡 상호작용 데이터	110,465 × 707,989	4,653,529
곡 특성 행렬 종류	곡 특성 행렬 크기	원소 개수
장르	707,989 × 708,211	1,415,978
발매연도	707,989 × 708,073	1,415,978
아티스트	707,989 × 828,273	1,415,978
장르 + 아티스트	707,989 × 828,495	2,123,967
장르 + 발매연도	707,989 × 708,295	2,123,967
발매연도 + 아티스트	707,989 × 828,345	2,123,964
아티스트 + 장르 + 발매연도	707,989 × 828,567	2,831,953

LightFM 학습을 위한 하이퍼파라미터는 <표 5>와 같다. LightFM을 바탕으로 곡 메타데이터 활용의 추천 성능 향상에 대한 효과를 베이스라인모델과 비교하기 위해서 차원 수는 ALS, BPR, Item2Vec과 같은 100차원으로 설정했다. 여기서 설정한 손실함수는 Weighted Approximate-Rank Pairwise(WARP) loss로 설정했다. WARP loss는 랭킹 오류가 나올 때까지 부정적 피드백 예시들을 반복적으로 샘플링해 긍정적 피드백을 가진 아이템의 랭킹을 최대화하는 손실함수이다(Kula, 2016). 또한, 해당 손실함수는 긍정적 상호작용 데이터(positive interactions)만 주어져 있고 추천 리스트 내 상위 순서를 최적화가 필요할 때 유용하다(Kula, 2016). 본 논문에서는 추천 대상 플레이리스트 ID 내에 수록될 확률이 높은 곡을 상위 100개로 추천하는 문제이므로 앞서 설명한 WARP 손실함수를 사용하였다. 이러한 요소는 LightFM과 비슷하게 아이템의 메타데이터를 활용하지만 손실함수로 Logloss를 활용하는 DeepFM(Guo et

al., 2017)과 차이점을 보인다. 따라서, 메타데이터를 활용하고 동시에 Top-N recommendation 문제처럼 추천 랭킹이 중요한 문제에서는 WARP loss를 학습에 활용하는 LightFM이 다른 방법론과 비교해서 가장 적합하다고 판단이 들어 해당 알고리즘을 활용하게 되었다. 이렇게 학습된 LightFM 모델의 유저와 아이템의 잠재요소 벡터를 바탕으로 추천 대상 플레이리스트 ID 별로 각 아이템 곡 마다 추천 확률 점수를 부여하였다. 그 다음, 추천 대상 플레이리스트 ID의 수록 곡을 제외하고 확률 점수가 높은 상위 100개 곡을 추천 리스트로 제공하였다.

〈표 5〉 LightFM 학습을 위해 설정된 하이퍼파라미터

하이퍼파라미터 명	설정 값
차원 수(no_components)	100
훈련 횟수(epochs)	200
아이템 L2 정규화(item_alpha)	1e-6
손실함수(loss)	warp

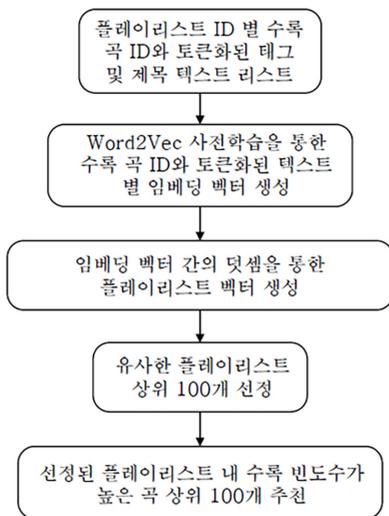
또한, 본 논문에서는 곡에 대한 메타데이터를 활용해 아이템 특성 행렬을 만들었지만, 유저에 해당하는 플레이리스트의 메타데이터는 태그와 제목과 같은 텍스트 데이터이므로 유저 특성 행렬은 따로 생성하지 않았다. 대신에, 추후에 언급될 Item2Vec에서 플레이리스트의 곡 ID 리스트와 전처리된 태그와 제목을 같이 학습해 플레이리스트 임베딩 벡터를 만들어 LightFM과 앙상블을 하는 방식으로 생성하지 않은 유저 특성 행렬에 대한 보완책을 마련하였다.

### 2.2.3 Item2Vec 학습 및 데이터 전처리

본 논문에서는 Item2Vec 기반 추천을 위해 각

플레이리스트 ID별로 수록 곡 ID 리스트와 토큰화된 텍스트 데이터를 입력 데이터로 사용해 Word2Vec 기법으로 각 곡 ID와 텍스트에 대한 임베딩 벡터를 생성하였다. 그 다음, 생성된 임베딩 벡터를 모두 더해 각 플레이리스트 ID에 해당하는 플레이리스트 벡터를 만들었다. 그리고 추천 대상 플레이리스트 ID의 플레이리스트 벡터와 유사한 임베딩 벡터를 가진 플레이리스트 상위 100개를 선정하고 이렇게 선정된 플레이리스트 안에 수록 빈도수가 높은 곡 순서대로 추천 대상 플레이리스트 ID의 수록 곡을 제외한 상위 100개를 추천하도록 하였다. 추천 대상 플레이리스트 ID와 다른 플레이리스트 간의 유사도는 코사인 유사도를 기반으로 계산했다.

앞서 설명한 Item2Vec의 데이터 전처리 및 추천 과정은 <그림 3>과 같다. 이러한 과정을 거친 Item2Vec 모델의 추천 결과는 추후에 LightFM 모델의 결과와 앙상블 과정을 거쳐 최종 추천 리스트를 생성한다.



<그림 3> Item2Vec 학습을 위한 데이터 전처리 및 추천 과정

Word2Vec 사전학습은 SGNS (Mikolov et al.,

2013) 기법을 활용하였다. 베이스라인 모델인 ALS, BPR, LightFM과 동일하게 임베딩 차원을 100개로 설정하였다. 베이스라인 모델로 활용될 Item2Vec 모델은 플레이리스트의 수록 곡 ID만을 학습한 모델이다. Word2Vec 사전학습을 위해 설정한 하이퍼파라미터는 <표 6>과 같다.

<표 6> Item2Vec 사전 학습을 위한 설정 하이퍼파라미터

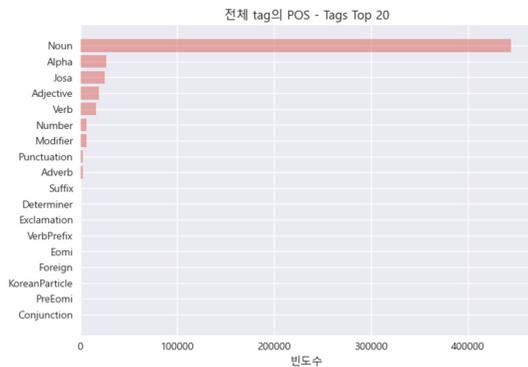
하이퍼파라미터 명	설정 값
임베딩 차원(vectore_size)	100
단어 최소 등장 허용 횟수 (min_count)	1
Skip-gram 사용 여부(sg)	1 (skip-gram 사용)
Hierarchical-softmax 사용 여부 (hs)	0 (negative sampling 사용)
Negative sampling 개수(negative)	50
윈도우 크기 (window)	300
훈련 횟수(epochs)	10

## 2.2.4 플레이리스트 제목과 태그 텍스트 데이터 사전 학습 및 전처리

플레이리스트의 제목과 태그를 통해 사용자가 어떠한 의도로 플레이리스트를 만들고 수록 곡을 정했는지 알 수 있다. 따라서 비슷한 의미를 가진 제목과 태그를 가진 플레이리스트들은 비슷한 곡으로 구성될 확률이 높다고 가정할 수 있다. 따라서 본 논문에서는 플레이리스트 안에 수록 곡이 없는 경우와 같은 콜드 스타트에 대응하기 위해 <그림 3>과 같이 태그 및 제목 기반으로 추천을 진행하였다. 하지만 이 때는 수록 곡이 없는 경우이므로 태그 및 제목 텍스트 데이터의 임베딩 벡터만을 생성해 추천에 활용하였다. 그러기 위해서는 플레이리스트의 메타데이터인 제목과 태그와 같은 텍스트 데이터에 대한 분석이 필요하다. 태그와 제목이 가지고 있는 특성을 파악하기 위해

KoNLPy라는 한국어 자연어처리 파이썬 라이브러리의 Twitter 패키지를 활용해 품사(POS) 분석한 결과는 아래 <그림 4>, <그림 5>와 같다.

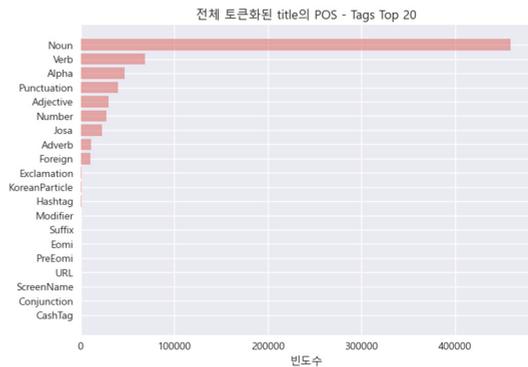
플레이리스트의 태그는 플레이리스트의 분위기와 의도를 표현할 수 있는 단어들의 리스트이므로 토큰화를 따로 진행하지 않았다. <그림 4>를 보면 플레이리스트 태그의 대부분이 명사(Noun)인 것을 확인할 수 있다.



<그림 4> 플레이리스트 태그의 품사(POS) 빈도수

플레이리스트 제목과 같은 경우에는 형태소 단위로 토큰화를 진행하고 불용어 제거 후 분리된 형태소의 태깅된 품사 빈도수를 분석했다. <그림 5>에서 볼 수 있듯이 불용어를 제거하면 태그와 비슷하게 대부분의 형태소가 명사(Noun)인 것을 확인할 수 있다. 따라서 본 논문에서는 플레이리스트 별로 태그와 불용어가 제거된 토큰화된 제목을 같은 리스트 안에 넣고 FastText를 기반으로 플레이리스트 별 태그와 제목을 기반으로 한 임베딩 벡터를 생성하였다. 여기서 FastText는 skipgram을 바탕으로 각 단어가 글자 단위 n-gram의 임베딩 벡터로 표현되는 기법이다(Bojanowski et al., 2017). 즉, 단어들은 n-gram 단위의 글자의 벡터 표현의 합으로 표현된다. 이러한 원리로 인해

FastText는 한 단어 안의 여러 내부 단어 즉, 서브워드(subword)를 고려하여 학습을 할 수 있게 되고 이는 훈련 데이터 셋에 존재하지 않은 단어(Out Of Vocabulary, OOV)와 빈도수가 적은 단어(rare word)에 대한 정확한 벡터 임베딩 값을 얻을 수 있게 한다(Bojanowski et al., 2017).



<그림 5> 토큰화된 플레이리스트 제목의 품사(POS) 빈도수

플레이리스트 제목과 태그를 기반으로 플레이리스트 임베딩 벡터를 만드는 데 FastText를 이용하는 이유는 크게 두 가지가 있다. 첫 번째로는 플레이리스트 태그와 제목의 길이가 길지가 않다. 본 연구의 경우, 플레이리스트 당 평균적으로 약 4개의 태그를 가지고 있었고, 토큰화된 플레이리스트 제목 역시 평균 6개 단어를 가지고 있었다. 또한, 플레이리스트의 태그와 토큰화된 제목을 합쳐도 평균 9개 단어를 가진다는 것을 알 수 있었다. 두 번째 이유는 <그림 4>와 <그림 6>을 보면 태그와 제목 데이터는 대부분 명사로 이루어졌고, 특히 태그는 단순히 플레이리스트의 분위기를 나타내는 단어의 나열인 것을 알 수 있다. 또한, 플레이리스트의 텍스트 특성 상 맞춤법이 맞지 않은 단어들이 많았다. 따라서, 문장 안의 문맥을 활용하여 단어들을 임베딩하는 방식인 트랜스포머 기반의

임베딩 기법을 활용하기 보다는 명사의 나열로 이루어지고 맞춤법이 정확하지 않은 단어로 구성된 짧은 텍스트에서는 글자 n-gram 단위로 학습하는 FastText가 더 정확한 임베딩 벡터 생성에 더 적합하다고 판단이 들었다. 따라서, 텍스트 데이터만을 임베딩할 때는 OOV와 빈도수가 적은 단어에 대응이 가능한 FastText를 주로 활용하였다.

또한, 일반적인 텍스트 데이터 전처리 과정과는 다르게 본 연구에서 활용하는 데이터 셋에서는 특수 문자와 이모티콘으로만 이루어진 데이터 셋이 다수 존재하여 특수문자 및 이모티콘은 삭제하지 않았다. 이모티콘 그 자체에 있는 의미를 학습하기 위해 이를 제거하지 않고 일반 단어로 취급해 사전 학습을 진행하였다. 특정 이모티콘과 비슷한 벡터를 가진 단어 혹은 이모티콘들을 뽑은 결과를 보면 해당 이모티콘과 연상되는 단어 혹은 이모티콘이 나온다는 것을 볼 수 있었다. 즉, 특정 이모티콘에도 의미가 있다는 것을 확인할 수 가 있다. 이러한 과정을 거치기 위하여 설정된 FastText 모델의 하이퍼파라미터는 <표 7>과 같다. Gensim(Rehurek, R., 2022) 라이브러리의 FastText 클래스는 n-gram 범위가 1인 기본 값만 지원하기 때문에 n-gram 범위는 1로 설정하였다.

<표 7> FastText 사전 학습을 위한 설정 하이퍼파라미터

하이퍼파라미터 명	설정 값
임베딩 차원(vector_size)	100
단어 최소 등장 허용 횟수 (min_count)	1
Skip-gram 사용 여부(sg)	1 (skip-gram 사용)
Hierarchical-softmax 사용 여부 (hs)	0 (negative sampling 사용)
Negative sampling 개수 (negative)	50
윈도우 크기 (window)	30
훈련 횟수(epochs)	10

## 2.3 플레이리스트 제목 & 태그 기반 추천

플레이리스트 안에 수록 곡이 없는 경우에 행해지는 플레이리스트 제목과 태그만을 기반으로 한 추천 과정은 Item2Vec 추천 과정인 <그림 3>과 비슷하다. 다만 다른 점은 <그림 3>에서 두 번째 단계에서 Word2Vec 대신에 FastText을 바탕으로 사전 학습을 하고 곡 ID는 제외하고 태그 및 제목 단어에 대해서만 임베딩 벡터를 생성한다는 점이다. Word2Vec과 FastText 중 어느 것이 추천에 효과가 있는지 실험하기 위해 추천성능을 비교하였다.

## 2.4 LightFM과 Item2Vec 앙상블 추천

플레이리스트에 수록 곡이 있는 경우, 각 곡의 메타데이터와 수록 곡 ID 데이터를 같이 학습에 활용한 LightFM 모델과 플레이리스트의 제목 및 태그와 수록 곡 ID를 같이 학습한 Item2Vec 모델이 생성한 추천 결과의 앙상블을 통해 추천 성능을 향상시키고자 하였다. 구체적으로 LightFM(아티스트)와 Item2Vec 모델에 공통된 결과가 있으면 정답 리스트에 포함될 확률이 높을 것이라고 가정하고 추천 리스트를 앞쪽으로 보내고 나머지는 Item2Vec보다 더 좋은 성능을 보인 LightFM 모델의 추천 결과로 총 100개를 채웠다. 이를 통해 구성 곡의 ID와 메타데이터만을 사용한 LightFM의 한계를 극복하고 구성 곡이 적은 상황에서의 콜드 스타트 상황에 대응하고자 하였다.

## 2.5 평가지표

본 논문에서는 각 추천 모델의 성능 평가를 위해 사용되는 평가지표는 Normalized Discounted Cumulative Gain(NDCG)이다. NDCG는 추천 모델을 순위(ranking)기반으로 평가할 때 흔히 쓰이

는 평가지표이다. NDCG의 공식은 아래와 같다.

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

NDCG에서 DCG는 아래 수식과 같이 각 추천 결과들의 관련성(relevance)에 랭킹의 순서에 따른 로그 값으로 나뉘 가중치를 줄여가는 방식이다. 즉, 추천 리스트의 결과가 정답 리스트에 안에 있고 앞쪽에 위치할수록 높은 점수를 받는 방식이다.

$$DCG_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)}$$

본 논문에서 관련성(relevance,  $rel_i$ )은 추천 리스트에 정답 리스트의 곡이 포함되면 1, 포함되지 않으면 0의 값을 가지는 이진 값으로 설정했다. 또한, NDCG에서 IDCG는 DCG가 가질 수 있는 이상적인 값, 즉 최댓값이다. 따라서 DCG를 IDCG로 나눈으로써 DCG가 이상적인 값인 IDCG와 같으면 최종적으로 NDCG는 1이라는 값을 가지고 추천 리스트에 정답 리스트와 일치하는 값이 없어 관련성이 모두 0이면 NDCG는 최종적으로 0이라는 값을 가진다.

### 3. 결과

본 절에서는 주요 실험 결과를 제시하고 분석했다. 모든 결과는 곡 100개를 추천한 결과에 대한 성능(NDCG@100)을 바탕으로 비교 분석한 것이다.

#### 3.1 메타데이터 활용에 따른 성능평가

본 항에서는 LightFM을 기반으로 어떠한 메타데이터를 활용해야 기존 MF 방법보다 향상된 추

천 성능을 낼 수 있는가?’의 연구 질문에 대한 분석 결과를 기술하였다. <표 4>에서 언급한 메타데이터 구성에 따른 아이템 특성 행렬과 플레이리스트-곡 상호작용 행렬을 바탕으로 학습한 LightFM 모델들의 <표 2>에 제시한 검증용 데이터에 대한 각 추천 성능 결과는 <표 8>과 같다. 각 LightFM 옆에 있는 괄호에는 학습에 사용된 <표 4>의 곡 특성 행렬 종류를 명시했다. <표 8>에는 MF에 해당하는 ALS와 BPR과 같은 베이스라인 모델의 결과도 함께 제시했다. 또한, <표 8>은 모델 기반 협업 필터링 모델에서 검증용 플레이리스트 데이터 안에 곡이 없는 경우에는 비개인화 추천 방식인 인기도 기반 추천을 진행한 것에 대한 결과이다. <표 8>의 인기도 기반 추천(POP)은 플레이리스트에 가장 많이 수록된 곡을 빈도수 순서대로 추천해 주는 방식이고 장르 기반 추천은 추천 대상 플레이리스트에 가장 많이 등장하는 장르에서 가장 많이 등장한 곡을 빈도수 순서대로 추천해 주는 방식이다.

<표 8> 기존 베이스라인과 활용 메타데이터에 따른 LightFM 모델의 성능표

알고리즘	NDCG@100
인기도 기반 추천(POP)	0.0169
장르 기반 추천	0.0409
BPR + POP	0.0433
ALS + POP	0.1149
LightFM(장르) + POP	0.1201
LightFM(발매연도) + POP	0.1231
LightFM(아티스트) + POP	0.1656
LightFM(장르, 발매연도) + POP	0.1149
LightFM(장르, 아티스트) + POP	0.1490
LightFM(발매연도, 아티스트) + POP	0.1558
LightFM(발매연도, 아티스트, 장르) + POP	0.1431

<표 8>에서 볼 수 있듯이 메타데이터를 활용한 모든 LightFM 모델이 비개인화 추천 방법인 인기도 기반과 장르 기반 추천 그리고 개인화 추천 방식 중에 MF에 해당하는 BPR보다 높은 성능을 보여줬다. 하지만 장르, 발매연도를 같이 활용한 LightFM 모델은 기존 개인화 추천 방식 중에서 MF에 해당하는 ALS와는 비슷한 성능을 보여줬고 그것을 제외한 모든 LightFM 모델이 ALS보다 더 높은 성능을 보여주었다. 또한, 아티스트 정보를 단독으로 사용한 LightFM 모델이 가장 높은 성능을 보여주었다. 이를 통해 곡 메타데이터 중에서 아티스트 정보가 추천 성능 향상에 가장 기여를 많이 한다는 것을 확인할 수 있었다. 즉, 음악 추천 도메인에서 아티스트라는 메타데이터가 제일 정확한 곡의 아이템 잠재요소 임베딩 벡터 값을 생성하는데 도움을 준다고 추정할 수 있었다.

### 3.2 수록 곡이 없을 때의 대응 방안

본 항에서는 ‘수록 곡이 없는 상황에 어떻게 대응할 수 있는가?’의 연구 질문에 대한 분석결과를 기술하였다. 플레이리스트 안에 곡이 아예 없는 경우에 단순히 인기도 기반 추천을 하게 될 경우에는 정확한 추천 결과를 얻기가 힘들 것이다. 따라서 이러한 상황에 대응하기 위해서 제3장 제2절 2.4에서 설명한 방식을 토대로 전처리가 된 플레이리스트의 제목과 태그를 기반으로 추천한 방식에 대한 전체적인 결과는 <표 9>와 같다. 여기서 플레이리스트 태그 및 제목을 임베딩하는 방식을 Word2Vec과 FastText로 달리해서 어떠한 방식이 더 정확한 추천을 해줄 수 있는지 알아보았다.

<표 9> 플레이리스트 태그 및 제목 기반 전체 추천 성능

플레이리스트 태그 및 제목 임베딩 기반 추천	NDCG@100
인기도 기반 추천(POP)	0.0169
장르 기반 추천	0.0409
Word2Vec	0.0887
FastText(자모 단위 토큰화)	0.0903
FastText	0.0922

검증용 데이터 셋에 대해서 추천을 진행할 때의 결과가 기존 비개인화 추천 방식인 인기도 기반 추천과 장르 기반 추천보다 더 좋은 성능을 보여주었다. 또한, 임베딩 기법을 Word2Vec과 FastText를 달리해서 추천을 한 결과, OOV와 빈도수가 적은 단어에 대응할 수 있는 FastText가 더 향상된 추천 성능을 보여주었다. 이는 플레이리스트 태그와 제목 같이 길이가 짧고 노이즈가 많은 데이터에서는 단어로 학습한 Word2Vec보다는 n-gram 단위의 글자로 학습하는 FastText가 더 정확한 임베딩 벡터 값을 생성한다고 추정할 수 있다.

추가적으로, 한글 텍스트 데이터는 hgtk(hangul-toolkit, <https://github.com/bluedisk/hangul-toolkit>) 라이브러리를 활용해 자모 단위로 토큰화가 가능하였다. 이러한 방식으로 전처리가 된 데이터를 바탕으로 FastText로 생성된 임베딩 벡터를 활용한 추천 방식은 Word2Vec보다는 향상된 추천 성능을 보여주었지만, 기존의 방식인 FastText 방식보다는 낮은 추천 성능을 보여준다는 것을 알 수 있었다. 즉, 자모 단위 토큰화가 추천 성능 향상에 기여하지는 못한다는 것을 확인할 수 있었다.

이와 비슷하게, 수록 곡이 없는 플레이리스트인 4606개 플레이리스트에 대해서도 태그 및 제목만을 이용해 추천을 할 때 Word2Vec보다는 FastText 기법을 활용한 추천 방법이 더 좋은 성능을 보여

줬다는 것을 <표 10>에서 확인할 수 있다.

<표 10> 플레이리스트 안에 수록 곡이 없을 때 활용되는 방법에 따른 추천 성능

플레이리스트 안에 수록 곡이 없을 경우 활용 추천 방법	NDCG@100
인기도 기반 추천(POP)	0.0199
Word2Vec	0.0996
<u>FastText</u>	<u>0.1039</u>

또한, 수록 곡이 없고 전처리가 된 태그 및 제목의 길이가 하나인 62개의 플레이리스트에 대해서도 Word2Vec보다 FastText 방식이 더 좋은 추천 성능을 보여준다는 것을 <표 11>에서 확인할 수 있었다. 이는 수록 곡이 없고 태그 및 제목의 데이터도 부족한 콜드 스타트 상황에서도 FastText를 활용한 추천 방식이 Word2Vec을 활용한 추천 방식보다 더 정확한 추천을 한다는 것을 의미한다.

또한, 수록 곡이 없고 전처리가 된 태그 및 제목의 길이가 하나인 62개의 플레이리스트에 대해서도 Word2Vec보다 FastText 방식이 더 좋은 추천 성능을 보여준다는 것을 <표 11>에서 확인할 수 있었다. 이는 수록 곡이 없고 태그 및 제목의 데이터도 부족한 콜드 스타트 상황에서도 FastText를 활용한 추천 방식이 Word2Vec을 활용한 추천 방식보다 더 정확한 추천을 한다는 것을 의미한다.

<표 11> 수록 곡이 없고 태그 및 제목의 길이가 하나일 때 활용되는 방법에 따른 추천 성능 비교

수록 곡이 없고 태그 및 제목의 길이가 하나인 경우의 활용 추천 방법	NDCG@100
인기도 기반 추천(POP)	0.0065
Word2Vec	0.0359
<u>FastText</u>	<u>0.0502</u>

종합해 보면, 수록 곡이 없는 플레이리스트에 대해서는 인기도 기반 추천 과 Word2Vec 대신 n-gram 단위의 글자를 기반으로 학습하는 FastText를 활용한다면 <표 10>, <표 11>과 같이 더 정확한 추천을 할 수 있다는 것을 확인할 수 있었다. 따라서 Switching 기법으로 구현한 하이브리드 추천 모델을 간단히 비교 분석한 결과(<표 12>), 수록 곡이 있는 경우에는 LightFM(아티스트) 모델로 추천을 진행하고 수록 곡이 없을 때에는 인기도 기반으로 추천한 모델 보다는 FastText 기반으로 추천을 진행한 하이브리드 모델의 전체적인 성능이 더 높은 것을 확인할 수 있다.

<표 12> 수록 곡이 없는 콜드 스타트 상황에서 인기도 기반과 FastText 기반으로 추천하는 하이브리드 추천 모델 성능 비교

알고리즘	NDCG@100
LightFM(아티스트) + POP	0.1656
LightFM(아티스트) + FastText	0.1823

### 3.3 수록 곡 개수가 적을 때의 대응 방안

본 항에서는 ‘어떠한 방법으로 수록 곡 개수가 적은 상황에서 성능 향상을 이룰 수 있는가?’의 연구 질문에 대한 분석결과를 기술하였다. 성능 향상을 위해 우선적으로 <표 8>에서 제일 좋은 성능을 보여준 LightFM(아티스트)를 기반으로 하이퍼파라미터 튜닝을 하였다. 이때, WARP로 모델을 학습할 때 쓸 수 있고 네거티브 샘플링 개수를 조절할 수 있는 파라미터인 max\_sampled를 100으로 설정하고 훈련 횟수(epoch)는 100으로 줄였다. 수록 곡이 있는 경우에는 앞서 하이퍼파라미터 튜닝된 LightFM(아티스트)을 활용하고 수록 곡이 없는 경우에는 인기도 기반으로 추

천한 하이브리드 모델에 대한 결과는 <표 13>에서 볼 수 있다.

<표 13> LightFM(아티스트) 하이퍼파라미터 조정에 따른 성능 결과

알고리즘	max_sampled	epochs	NDCG@100
LightFM(아티스트) + POP	10	200	0.1656
	100	100	0.1856

추가적으로, 전처리된 플레이리스트 태그 및 제목 리스트를 곡 ID와 함께 같은 리스트 안에 넣어서 Item2Vec으로 사전학습을 거쳐 추천을 진행하였다. 이것 역시 수록 곡이 있는 경우에는 Item2Vec을 활용하고 수록 곡이 없는 경우에는 인기도 기반으로 추천한 하이브리드 모델이다. 그 결과, <표 14>와 같은 추천 성능을 얻을 수 있었다.

<표 14> 수록 곡 ID와 플레이리스트의 태그 및 제목을 활용한 Item2Vec 성능 결과

알고리즘	활용 데이터	NDCG@100
Item2Vec + POP	수록 곡 ID, 플레이리스트 태그 및 제목	0.1774

<표 13>과 <표 14>에 제시된 LightFM(아티스트)와 Item2Vec 모델에 대한 결과에 대한 앙상블을 통해 플레이리스트 안에 수록 곡이 있는 경우, 추천 성능을 향상시켰다. 구체적으로 LightFM(아티스트)와 Item2Vec 모델에 공통된 결과가 있으면 <표 2>의 정답 데이터에 포함될 확률이 높을 것이라고 가정하고 추천 리스트를 앞쪽으로 보내고 나머지는 Item2Vec보다 더 좋은 성능을 보인 LightFM(아티스트)의 추천 결과로 총 100개를 채웠다.

결과적으로, 평균적으로 46개의 수록 곡을 가

진 다른 플레이리스트에 비해서 수록 곡의 개수가 1개 이상이고 5개 이하로 적은 콜드 스타트 상황(<그림 6>)에서 제안된 최종 앙상블 추천 모델이 수록 곡 ID와 곡의 메타데이터만을 활용한 기존 베이스라인 모델보다 더 좋은 성능을 보인 것을 <표 15>에서 확인할 수 있다.

tags	id	playlist_title	songs	like_cnt	updt_date
2	[ ]	요진이네 방(음)	[690947, 156049, 18368, 135345]	5	2009-06-07 11:30:36.000
12	[ 인기동화 ]	동화로 배워봐 유아 영어 동화	[328899, 611802, 361943, 435829, 391345]	3	2020-03-05 14:39:00.000
41	[ ]	여름에 들으면 시원해지는 노래	[507194, 388924]	5	2015-07-20 01:53:03.000
86	[ 취미 ]	#POP MUSIC	[186500, 633618, 477206, 596968]	636	2018-06-31 06:41:53.000
106	[ 추억 ]	DJ 미러볼 11회 리믹스	[271259, 556021, 71980, 218634, 365463]	31	2014-09-01 14:03:33.000
22961	[ ]	클러식 최신곡 모음집 (2015년 11월 30일)	[353629, 123697, 58314, 233712]	0	2015-12-01 18:00:25.000
22966	[ ]	라디오 696	[288958, 464767, 11811, 606057]	8	2010-06-10 09:30:58.000
22980	[ 단스 ]	맑은가을하늘 맑기 좋은 단스	[365398, 191431, 409908, 319354, 515356]	4	2017-10-12 07:07:51.000
22993	[ ]	김구 김은 우물.	[639720, 1508, 367433, 429971]	4	2008-06-25 23:04:26.000
23014	[ ]	라디오 389	[243082, 383812, 218279, 297891]	7	2009-10-01 13:58:17.000

1273 rows \* 6 columns

<그림 6> 수록 곡 개수가 적은 플레이리스트

<표 15> 수록 곡이 적은 콜드 스타트 상황에서의 성능 결과

알고리즘	NDCG@100
BPR (Rendle et al., 2009)	0.0761
ALS (Hu et al., 2008)	0.1137
Item2Vec (Barkan et al., 2016)	0.1734
LightFM(아티스트) (Kula et al., 2015)	0.1745
Ensemble(LightFM(아티스트), Item2Vec)	0.1923

즉, 플레이리스트 안에 수록 곡 개수가 적을 때의 콜드 스타트 상황에서는 단순히 수록 곡 ID와 곡의 메타데이터만을 사용하는 것보다는 각 곡과 각 플레이리스트의 메타데이터를 같이 활용하는 것이 더 좋은 성능을 낼 수 있다는 것을 알 수 있었다.

최종적으로, 본 논문에서 제안된 하이브리드 모델(플레이리스트 안에 수록 곡이 있는 경우에는 LightFM과 Item2Vec을 앙상블한 모델 기반으로

추천하고 수록 곡이 없으면 FastText 기반으로 추천하는 하이브리드 모델이 수록 곡이 있을 때 ALS, BPR, Item2Vec, LightFM 베이스라인 모델들 중에 하나를 활용하고 수록 곡이 없을 때 인기도 기반 추천하는 하이브리드 모델들보다 전체적으로 더 높은 추천 성능 결과를 얻은 것을 <표 16>에서 확인할 수 있다. 여기서 베이스라인 모델에 해당하는 Item2Vec은 각 수록 곡 ID만을 사전학습해 생성된 임베딩 벡터를 사용한 추천 모델이다.

<표 16> 최종 모델 전체 성능 결과

알고리즘	NDCG@100
BPR + POP	0.0435
ALS + POP	0.1149
Item2Vec + POP	0.1784
LightFM(아티스트) + POP	0.1656
Ensemble(LightFM(아티스트), Item2Vec) + FastText (본 논문에서 제안한 최종 하이브리드 추천 모델)	0.2072

이로써 기존 LightFM 모델에서는 플레이리스트의 메타데이터인 태그와 제목과 같은 텍스트 데이터를 플레이리스트의 특성으로 넣지 않는 한계가 있었지만, 플레이리스트의 태그와 제목을 같이 학습한 Item2Vec과의 앙상블을 통해 이러한 한계점에 대한 보완책을 마련할 수 있었다. 또한, 플레이리스트 안에 수록 곡이 아예 없고 태그 및 제목 텍스트 데이터가 적거나 수록 곡의 개수가 적을 때와 같은 다양한 콜드 스타트 상황에서 대응이 가능한 하이브리드 모델을 구축할 수 있었다.

## 4. 고찰

### 4.1 의의

본 논문에서는 다음과 같은 의의를 가진다. 첫째, 음악 추천이라는 특정 도메인에 추천 성능 향상에 가장 도움이 되는 음악 아이템의 메타데이터 구성이 무엇인지를 실험 분석을 통해 증명했다. 제2장 제4절에 언급한 음악 추천 시스템 선행연구에는 단지 특정 음악 메타데이터 중에 아티스트 정보만을 사용했다는 언급만 있었을 뿐 어떠한 메타데이터가 음악 추천 성능 향상에 가장 기여를 많이 하는지에 대한 비교 분석은 하지 않았다. 예를 들어, Yang et al.(2018)은 음악 아이템의 메타데이터인 아티스트와 플레이리스트의 제목을 활용해 오코인코더 기반 음악 추천 시스템을 제안했다. 또한, Vasile et al.(2016)은 기존 Item2Vec에서 확장해서 아이템의 메타데이터를 활용해 아이템의 임베딩 벡터를 생성하는 Meta-Prod2Vec을 제안했지만, Yang et al.(2018)과 비슷하게 곡의 메타데이터인 아티스트 정보만을 사용했다. 따라서 본 논문에서는 기존 연구와는 다르게 기존 MF에 유저와 아이템의 메타데이터를 기반으로 잠재요소 벡터를 생성해 추천해줄 수 있는 LightFM 알고리즘을 통해 곡 아이템의 메타데이터의 구성을 달리해 아이템 특성 행렬을 만들어 각 구성에 대한 성능을 비교 분석해 음악 추천이라는 도메인에서 추천 성능 향상에 가장 기여를 많이 하는 곡 아이템의 메타데이터가 무엇인지 탐색했다. 결과적으로, 본 논문에서는 곡 아이템의 메타데이터 중에 아티스트 정보를 단독으로 사용했을 때 제일 높은 성능을 보여줬다는 것을 입증했다. 따라서 추후에 등장할 곡 아이템의 아티스트라는 메타데이터와 유저-

사용자 상호작용 데이터를 동시에 학습으로 활용하는 추천 시스템 연구의 근거로 본 연구를 활용할 수 있다는 기대효과가 있다.

두 번째, 기존 협업 필터링에서 발생할 수 있는 콜드 스타트 문제를 해결하기 위해서 멜론 서비스 도메인에 맞는 하이브리드 추천 전략을 제안했다. 기존 협업 필터링 기반 추천 시스템은 콘텐츠 기반 필터링보다 성능이 좋다는 장점이 있지만 사용자-아이템 상호작용 데이터를 기반으로 하기 때문에 아이템 사용 이력이 없거나 적은 사용자에 대한 추천 성능이 떨어지는 콜드 스타트 문제를 갖고 있다. Vasile et al.(2016)의 연구에서는 아이템의 메타데이터를 활용해 상호작용 데이터가 적을 때 발생하는 콜드 스타트 상황은 대응할 수 있었지만, 본 논문처럼 상호작용이 아예 없는 상황에서의 대응책은 마련하지 않았다. 따라서 본 논문에서는 이러한 문제를 해결하기 위해서 Burke et al.(2002)이 분류한 7가지 하이브리드 추천 방법 중 Switching 방법을 사용했다. 구체적으로 플레이리스트에 수록 곡이 있는 경우에는 LightFM과 Item2Vec의 앙상블 모델을 활용하고 수록 곡이 없는 경우에는 플레이리스트의 태그와 제목만을 이용해 추천을 해주는 방식을 제안했다. 이는 플레이리스트에 수록 곡이 없을 때 단순히 인기도 기반으로 추천하는 방식보다 향상된 성능을 보여줬다. 또한, 플레이리스트의 태그와 제목 기반으로 임베딩 벡터를 생성하기 위한 방법으로 Word2Vec과 FastText를 활용한 추천을 구현해 비교했다. 그 결과, n-gram 단위의 FastText를 활용한 추천이 더 성능이 좋다는 것을 실험을 통해 입증했다. 또한, FastText를 활용한 추천 방법이 플레이리스트 안에 수록 곡이 없고 태그 및 제목의 길이가 하나인 콜드 스타트 상황에서도 Word2Vec보다 더 좋은 성능을

보여줬다. 이를 통해, FastText를 활용한 추천을 바탕으로 다양한 콜드 스타트 상황에 대응이 가능한 하이브리드 추천 모델을 구현할 수 있었다.

세 번째, 곡 아이템의 메타데이터 뿐만 아니라 텍스트 데이터로 이루어진 플레이리스트의 메타데이터를 함께 활용한 방안을 제시했다. 특히, 수록 곡 ID와 곡 아이템의 메타데이터만을 활용한 LightFM의 단점을 앞서 전처리한 플레이리스트의 태그와 제목과 같은 텍스트 데이터와 수록 곡 ID를 같이 학습한 Item2Vec의 결과를 앙상블에 활용함으로써 보완할 수 있었다. 결과적으로, 플레이리스트의 수록 곡 목록과 곡과 플레이리스트의 메타데이터를 동시에 활용하는 방안을 제시할 수 있었다. 또한, 수록 곡 ID와 곡의 메타데이터를 활용한 LightFM 모델과 수록 곡 ID와 플레이리스트의 태그 및 제목을 같이 학습한 Item2Vec의 모델이 생성한 결과의 앙상블을 통해 플레이리스트의 수록 곡 개수가 적은 상황의 콜드 스타트에서도 대응할 수 있는 하이브리드 추천 모델을 구축할 수 있었다.

## 4.2 한계점 및 추후 연구 과제

본 논문의 한계점으로는 곡 아이템의 메타데이터와 플레이리스트의 제목과 태그와 같은 텍스트 데이터를 독립적으로 학습을 하고 추천에 활용됐다는 점이다. 이와 다르게 플레이리스트의 태그와 제목 그리고 플레이리스트 안에 있는 수록 곡의 메타데이터를 동시에 같이 학습할 수 있다면 플레이리스트의 더 정확한 특성을 파악하고 더 정교한 추천이 가능할 것이라고 기대된다.

또한, 본 논문에서는 협업 필터링의 단점을 보완하기 위한 Switching을 전략을 바탕으로 한 하이브리드 추천 전략에 방안을 제시했지만, 이는

상황에 따라서 서로 다른 모델을 만들어야 한다는 단점이 있다. 한가지 모델로 콜드 스타트 상황을 해결할 수 있다면 더 효율적으로 추천 시스템 모델을 설계할 수 있을 것이다. 따라서 추후 연구에서는 다양한 콜드 스타트 상황을 대응할 수 있는 한 가지 모델을 구축하는 것에 대한 연구가 필요해 보인다.

## 5. 결론

본 연구는 음악 추천 도메인에서 아이템에 해당하는 음악의 어떠한 메타데이터가 추천 향상에 가장 기여를 많이 하는지 탐색하기 위해 LightFM의 아이템 특성 행렬의 구성을 달리해 알아보고 상황에 맞는 추천 시스템을 달리하는 하이브리드 추천 시스템 전략을 제시하였다.

결과적으로 NDCG@100을 기준으로 아티스트 단일 정보를 기반으로 생성된 아이템 특성 벡터를 학습한 LightFM 모델의 결과가 다른 메타데이터를 구성해 생성된 특성 벡터를 학습한 LightFM 모델들과 비교해서 성능이 가장 높게 나타났다. 따라서 아티스트 정보가 정확한 음악 아이템 잠재 특성 벡터를 만드는데 제일 도움이 되는 메타데이터라고 할 수 있다. 또한, 아티스트라는 메타데이터만으로 음악의 장르 혹은 발매연도가 표현하는 아이템의 잠재 요소도 표현할 수 있다고 추정해볼 수 있다.

또한, 본 연구에서는 플레이리스트 안에 수록곡이 없어 플레이리스트와 곡 간에 상호작용이 없는 경우를 대응할 수 있는 방안으로 인기도 추천 보다는 비슷한 의미를 가진 제목 혹은 태그를 기반으로 하는 것이 더 정확한 추천 결과를 낼 수 있다는 것을 입증할 수 있었다. 구체적으로

플레이리스트 태그와 제목의 의미를 학습할 때는 단어 단위로 학습하는 Word2Vec보다는 n-gram 단위의 글자로 학습하는 FastText가 더 효율적이라는 것을 증명할 수 있었다.

마지막으로, 수록 곡 ID와 아티스트 메타데이터를 활용한 LightFM 모델과 플레이리스트의 태그와 제목 그리고 수록 곡 ID를 같이 학습한 Item2Vec 모델의 결과를 조합한 앙상블 모델이 기존 베이스라인 모델보다 향상된 성능을 보여주었다. 또한, 이를 통해 플레이리스트 안에 수록 곡 개수가 적은 콜드 스타트 상황에서도 좋은 성능을 보여주었다. 결과적으로 다양한 콜드 스타트 상황에 대응이 가능한 하이브리드 추천 모델을 제시할 수 있었다.

## 참고문헌(References)

### [국내 문헌]

- 김동환. (2023). 언어모델을 활용한 콘텐츠 메타데이터 기반 유사 콘텐츠 추천 모델. *지능정보연구*, 29(1), 27-40.
- 장동수, 이청용, 김재경. (2023). 딥러닝 기반 온라인 리뷰의 언어학적 특성을 활용한 추천 시스템 성능 향상에 관한 연구. *지능정보연구*, 29(1), 41-63.
- 홍태호, 홍준우, 김은미, 김민수. (2022). 영화 리뷰의 상품 속성과 고객 속성을 통합한 지능형 추천시스템. *지능정보연구*, 28(2), 1-18.

### [국외 문헌]

- Barkan, O., & Koenigstein, N. (2016, September). Item2vec: neural item embedding for collaborative filtering. In 2016 IEEE 26th International Workshop on Machine Learning for Signal

- Processing (MLSP) (pp. 1-6). IEEE.
- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5, 135-146.
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4), 331-370.
- Ferraro, A., Kim, Y., Lee, S., Kim, B., Jo, N., Lim, S., Lim, S., Jang, J., Kim, S., Serra, X., & Bogdanov, D. (2021). Melon Playlist Dataset: A Public Dataset for Audio-Based Playlist Generation and Music Tagging. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 536-540.
- Frederickson, B. (2017). AlternatingLeastSquares. Implicit. Retrieved December 6, 2022, from [https://benfred.github.io/implicit/api/models/cp\\_u/als.html](https://benfred.github.io/implicit/api/models/cp_u/als.html).
- Guo, H., Tang, R., Ye, Y., Li, Z., He, X., (2017, March). DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. *arXiv preprint arXiv:1703.04247*.
- Hu, Y., Koren, Y., & Volinsky, C. (2008, December). Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining* (pp. 263-272). Ieee.
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37.
- Kula, M. (2015). Metadata embeddings for user and item cold-start recommendations. *arXiv preprint arXiv:1507.08439*.
- Kula, M. (2016). The LightFM model class. LightFM. Retrieved December 6, 2022, from <https://making.lyst.com/lightfm/docs/index.html>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Rehurek, R. (2022). FastText model. Gensim Topic modelling for humans, Retrieved December 6, 2022, from <https://radimrehurek.com/gensim/models/fasttext.html>
- Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.
- Vasile, F., Smirnova, E., & Conneau, A. (2016, September). Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 225-232).
- Vasile, F., Smirnova, E., & Conneau, A. (2016, September). Meta-prod2vec: Product embeddings using side-information for recommendation. In *Proceedings of the 10th ACM conference on recommender systems* (pp. 225-232).
- Virtanen, P., Gommers, R., Oliphant, T.E. et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods*, 17, 261-272. <https://doi.org/10.1038/s41592-019-0686-2>
- Yang, H., Jeong, Y., Choi, M., & Lee, J. (2018). Mmcf: Multimodal collaborative filtering for automatic playlist continuation. In *Proceedings of the ACM Recommender Systems Challenge 2018* (pp. 1-6).

## Abstract

# Research on hybrid music recommendation system using metadata of music tracks and playlists

Hyun Tae Lee\* · Gyoo Gun Lim\*\*

Recommendation system plays a significant role on relieving difficulties of selecting information among rapidly increasing amount of information caused by the development of the Internet and on efficiently displaying information that fits individual personal interest. In particular, without the help of recommendation system, E-commerce and OTT companies cannot overcome the long-tail phenomenon, a phenomenon in which only popular products are consumed, as the number of products and contents are rapidly increasing. Therefore, the research on recommendation systems is being actively conducted to overcome the phenomenon and to provide information or contents that are aligned with users' individual interests, in order to induce customers to consume various products or contents.

Usually, collaborative filtering which utilizes users' historical behavioral data shows better performance than contents-based filtering which utilizes users' preferred contents. However, collaborative filtering can suffer from cold-start problem which occurs when there is lack of users' historical behavioral data.

In this paper, hybrid music recommendation system, which can solve cold-start problem, is proposed based on the playlist data of Melon music streaming service that is given by Kakao Arena for music playlist continuation competition. The goal of this research is to use music tracks, that are included in the playlists, and metadata of music tracks and playlists in order to predict other music tracks when the half or whole of the tracks are masked. Therefore, two different recommendation procedures were conducted depending on the two different situations.

When music tracks are included in the playlist, LightFM is used in order to utilize the music track list of the playlists and metadata of each music tracks. Then, the result of Item2Vec model, which uses vector embeddings of music tracks, tags and titles for recommendation, is combined with the result of LightFM model to create final recommendation list. When there are no music tracks available in the

---

\* Business Informatics, Hanyang University

\*\* Corresponding Author: Gyoo Gun Lim

School of Business, Hanyang University

#609, Business School, Hanyang University, Wangsimri-ro, Seongdong-gu, Seoul, Republic of Korea

Tel: +82-2-2220-2593, E-mail: gglim@hanyang.ac.kr

playlists but only playlists' tags and titles are available, recommendation was made by finding similar playlists based on playlists vectors which was made by the aggregation of FastText pre-trained embedding vectors of tags and titles of each playlists.

As a result, not only cold-start problem can be resolved, but also achieved better performance than ALS, BPR and Item2Vec by using the metadata of both music tracks and playlists. In addition, it was found that the LightFM model, which uses only artist information as an item feature, shows the best performance compared to other LightFM models which use other item features of music tracks.

**Key Words** : Hybrid recommendation system, Cold-start, LightFM, Item2Vec, FastText, Pre-training

Received : May 7, 2023 Revised : July 28, 2023 Accepted : August 25, 2023

Corresponding Author : Gyoo Gun Lim

## 저자 소개



### 이현태

한양대학교 경영대학 비즈니스인포매틱스학과 석사학위를 취득했다. 주요 관심분야는 감성분석, 자연어처리 그리고 추천 시스템이다.



### 임규건

한양대학교 경영대학 임규건 교수는 KAIST 전산학 학사, POSTECH 컴퓨터 석사, KAIST 경영공학 박사학위를 취득하였고, 삼성전자, KT, 국제전자상거래 연구센터 (ICEC) 연구위원, 세종대학교 경영학과 교수를 역임하였다. 관심 분야는 혁신비즈니스 모델, IT서비스 혁신, 인공지능과 경영, e-Business 등이며, 2018년 IT서비스 우수연구인상을, 2009년 IT Innovation 유공자 지식경제부 장관 표창과 2007년 SW산업발전 유공자 정통부 장관 표창을 수여하였다. 주요 저서로는 ‘경영을 위한 정보기술’, ‘e-비즈니스 경영’, ‘디지털경제시대의 경영정보시스템’ 등 전문서적과 다수의 논문과 특허가 있다. 또한, 아시아최초 상용인터넷인 KORNET 상용화, 중국 Shanghai Telecom SI사업전략, 한국영화 기술 로드맵, KTI 사업전략, 나라장터 (G2B) 효과평가, 행정정보화(G4C) 성과분석, 국가정보보호지수개발, 국방정보화수준평가모형, IT혁신인력양성 종합대책, 국가디지털식별체계(UCI), 저작권정품인증제도, SW사업자신고 제도개선, SW기술자신고제도 개선 등 다양한 IT혁신분야의 프로젝트를 수행하였다.