데이터분석캡스톤디자인

11주차 수행보고

Khupid 조

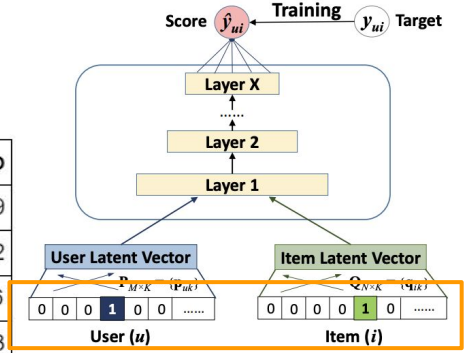
산업경영공학과 김동혁
관광학과 류연주
산업경영공학과 유정수

10-11주차 이슈

	group_5	group_0	group_1	group_2	group_3	group_4	group_6	group_7	group_8	group_9	group_10	group_11
0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	2	0	2
3	2	4	15	1	0	0	5	13	0	21	0	3
4	0	0	1	0	0	0	0	0	3	1	0	0
5	1	0	3	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	1	1	1	0	0	1	3	0	1	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0

?

	userID	itemID	rating	timestamp
0	196	242	3.0	881250949
1	186	302	3.0	891717742
2	22	377	1.0	878887116
3	244	51	2.0	880606923
4	166	346	1.0	886397596



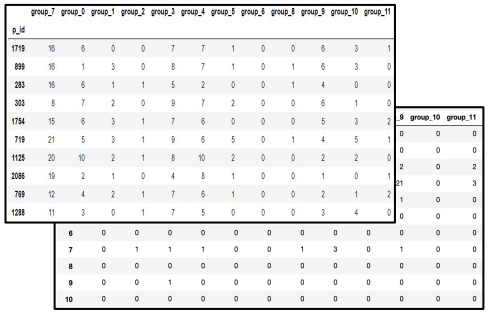
➔ 그동안 만든 형용사 메트릭스를 어떻게 NCF에 적용시켜야 하나 고민
➔ 평점 기반의 NCF 모델과 content-based 모델을 혼동해서 생긴 이슈
➔ 두 모델을 합친 **hybrid 방법**을 사용하기로 결정
➔ 모델에 필요한 matrix 추가 생성

모델 적용 방향 수정

Content-based Filtering

유저 특성과 항목 특성 사이의 코사인 유사도를 구한 후

가장 유사한 항목을 정렬 후 k를 고름

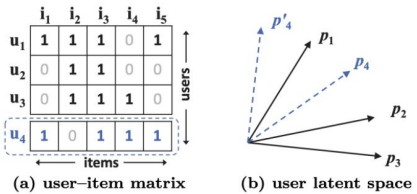


Item matrix & user matrix

Neural Collaborative Filtering

microsoft에서 만든 NCF 라이브러리를 활용

split method와 parameter 값을 조정해가며 precision@k 와 recall@k를 높임



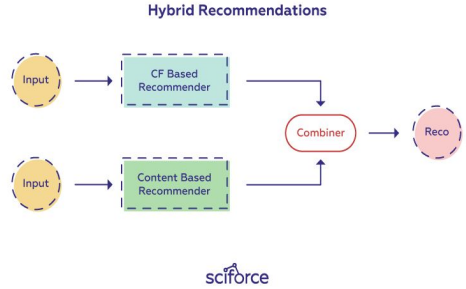
(a) user–item matrix (b) user latent space

Hybrid Methods

상위 k개의 content-based filtering 추천 결과를 가져오고,

상위 k개의 neural-collaborative filtering 추천 결과를 가져와서

두 모형 결과를 곱해서 최종 추천 아이템 선정



NCF 모델 테스트

$$\textbf{Precision@K} = \frac{\text{실제로 User가 관심있는 Item 수}}{\text{알고리즘이 추천한 top K개의 Item 수}}$$

$$\textbf{Recall@K} = \frac{\text{User가 관심있는 Item 중 top K개의 Item 수}}{\text{실제 User가 관심있는 Item 수}}$$

ex) 알고리즘 추천 {1, 3, 5, 7, 9}
실제 사용자 선호 {1, 2, 5}

precision@k = 2/5 = 0.4

recall@k = 2/3 = 0.66

Data per User (K=10)	갈 곳	볼 곳	먹을 곳
>=1	Total data : 60701 MAP: 0.414834 NDCG: 0.485684 Precision@K: 0.096972 Recall@K: 0.628179	Total data : 29560 MAP: 0.009116 NDCG: 0.014668 Precision@K: 0.005237 Recall@K: 0.025850	Total data : 19659 MAP: 0.006007 NDCG: 0.011613 Precision@K: 0.005315 Recall@K: 0.019388
>=10	Total data : 17550 MAP: 0.285411 NDCG: 0.421233 Precision@K: 0.214122 Recall@K: 0.460663	Total data : 18866 MAP: 0.031093 NDCG: 0.081540 Precision@K: 0.060079 Recall@K: 0.065089	Total data : 9292 MAP: 0.010099 NDCG: 0.024642 Precision@K: 0.017080 Recall@K: 0.027271

12주차에 할 일

1. precision@K, recall@K가 낮은 이유가 데이터가 적은 탓인가?

- 데이터 추가 수집?
- epoch, batch_size, layer size 등 모델 파라미터 수정

2. 최종 모델 테스트

- content-based model 구축
- hybrid method 적용 후 성능 비교

Algorithms

The table below lists the recommender algorithms currently available in the repository. Notebooks are linked under the Environment column when different implementations are available.

Algorithm	Environment	Type	Description
Alternating Least Squares (ALS)	PySpark	Collaborative Filtering	Matrix factorization algorithm for explicit or implicit feedback in large datasets, optimized by Spark MLLib for scalability and distributed computing capability
Attentive Asynchronous Singular Value Decomposition (A2SVD)[*]	Python CPU / Python GPU	Collaborative Filtering	Sequential-based algorithm that aims to capture both long and short-term user preferences using attention mechanism
Cornac/Bayesian Personalized Ranking (BPR)	Python CPU	Collaborative Filtering	Matrix factorization algorithm for predicting item ranking with implicit feedback
Convolutional Sequence Embedding Recommendation (Caser)	Python CPU / Python GPU	Collaborative Filtering	Algorithm based on convolutions that aims to capture both user's general preferences and sequential patterns
Deep Knowledge-Aware Network (DKN)[*]	Python CPU / Python GPU	Content-Based Filtering	Deep learning algorithm incorporating a knowledge graph and article embeddings to provide powerful news or article recommendations
Extreme Deep Factorization Machine (xDeepFM)[*]	Python CPU / Python GPU	Hybrid	Deep learning based algorithm for implicit and explicit feedback with user/item features
FastAI Embedding Dot Bias (FAST)	Python CPU / Python GPU	Collaborative Filtering	General purpose algorithm with embeddings and biases for users and items
LightFM/Hybrid Matrix Factorization	Python CPU	Hybrid	Hybrid matrix factorization algorithm for both implicit and explicit feedbacks
LightGBM/Gradient Boosting Tree[*]	Python CPU / PySpark	Content-Based Filtering	Gradient Boosting Tree algorithm for fast training and low memory usage in content-based problems
GRU4Rec	Python CPU / Python GPU	Collaborative Filtering	Sequential-based algorithm that aims to capture both long and short-term user preferences using recurrent neural networks
Neural Recommendation with Long- and Short-term User Representations (LSTUR)[*]	Python CPU / Python GPU	Content-Based Filtering	Neural recommendation algorithm with long- and short-term user interest modeling
Neural Recommendation with Attentive Multi-View Learning (NAML)[*]	Python CPU / Python GPU	Content-Based Filtering	Neural recommendation algorithm with attentive multi-view learning
Neural Collaborative Filtering (NCF)	Python CPU / Python GPU	Collaborative Filtering	Deep learning algorithm with enhanced performance for implicit feedback
Neural Recommendation with Personalized Attention (NPA)[*]	Python CPU / Python GPU	Content-Based Filtering	Neural recommendation algorithm with personalized attention network
Neural Recommendation with Multi-Head Self-Attention (NRMS)[*]	Python CPU / Python GPU	Content-Based Filtering	Neural recommendation algorithm with multi-head self-attention
Restricted Boltzmann Machines (RBM)	Python CPU / Python GPU	Collaborative Filtering	Neural network based algorithm for learning the underlying probability distribution for explicit or implicit feedback
Riemannian Low-rank Matrix Completion (RLRMC)[*]	Python CPU	Collaborative Filtering	Matrix factorization algorithm using Riemannian conjugate gradients optimization with small memory consumption.
Simple Algorithm for Recommendation (SAR)[*]	Python CPU	Collaborative Filtering	Similarity-based algorithm for implicit feedback dataset
Short-term and Long-term preference Integrated Recommender (SLi-Rec)[*]	Python CPU / Python GPU	Collaborative Filtering	Sequential-based algorithm that aims to capture both long and short-term user preferences using attention mechanism, a time-aware controller and a content-aware controller
Surprise/Singular Value Decomposition (SVD)	Python CPU	Collaborative Filtering	Matrix factorization algorithm for predicting explicit rating feedback in datasets that are not very large
Term Frequency - Inverse Document Frequency (TF-IDF)	Python CPU	Content-Based Filtering	Simple similarity-based algorithm for content-based recommendations with text datasets
Vowpal Wabbit Family (VW)[*]	Python CPU (online training)	Content-Based Filtering	Fast online learning algorithms, great for scenarios where user features / context are constantly changing
Wide and Deep	Python CPU / Python GPU	Hybrid	Deep learning algorithm that can memorize feature interactions and generalize user features
xLearn/Factorization Machine (FM) & Field-Aware FM (FFM)	Python CPU	Content-Based Filtering	Quick and memory efficient algorithm to predict labels with user/item features