

# 다중 클래스 환경의 추천 시스템을 위한 지식 증류 기법들의 비교 분석

김지연, 배홍균, 김상욱<sup>1</sup>  
한양대학교 컴퓨터소프트웨어학과  
{jiyeon7, hongkyun, wook}@hanyang.ac.kr

## Knowledge Distillation for Recommender Systems in Multi-Class Settings: Methods and Evaluation

Jiyeon Kim, Hong-Kyun Bae, Sang-Wook Kim  
Dept. of Computer Science, Hanyang University

### 요약

추천 시스템에서 사용되는 피드백은 단일 클래스와 다중 클래스 피드백으로 구분할 수 있다. 추천 시스템을 위한 지식 증류 기법들은 단일 클래스 환경에서 주로 연구되어 왔다. 우리는 다중 클래스 환경에서 또한 추천 시스템을 위한 최신 지식 증류 기법들이 효과적인지에 대해 알아보고자 하며, 해당 방법들 간의 추천 정확도를 비교해보고자 한다. 추천 시스템에서 보편적으로 사용되는 데이터 셋들을 기반으로 한 실험들을 통해 추천 시스템을 위한 지식 증류 기법들은 같은 조건의 기본적인 추천 시스템에 비해 정확도가 최대 193%까지 개선되는 것을 확인했다.

### 1. 서론

최근 추천 시스템 도메인에서 사용되는 머신러닝/딥러닝 모델들의 크기가 커짐에 따라 [1,2], 지식 증류 기법 적용 관련 연구들이 활발히 진행되고 있다. 지식 증류 기법은 모델 압축 기술 중의 하나로, 크기가 큰 교사 모델을 활용해서 얻은 추가적인 지식을 활용해서 크기가 작은 학생 모델이 더 짧은 시간 안에 교사 모델과 비슷한 정확도를 보일 수 있도록 한다 [3-6]. 추천 시스템을 위해 제안된 기존의 지식 증류 기법 방법들로는, Ranking Distillation (RD), Collaborative Distillation (CD), Distillation Experts - Relaxed Ranking Distillation (DE-RRD), Bidirectional Distillation (BD), Topology Distillation (TD) [7-11] 등이 존재한다.

하지만 이러한 기존 방법들은 주로 단일 클래스 환경을 타겟으로 제안됐었으며, 단일 클래스 피드백을 이용하는 환경 안에서만 서로들 간의 추천 정확도를 비교해왔다. 단일 클래스 피드백은 사용자의 행위에 대한 정보로서, 클릭했는지 안 했는지 등과 같이 아이템 사용 여부를 알 수 있는 데이터이다. 반면, 다중 클래스 피드백은 사용자가 아이템을 사용한 후, 어느 정도로 만족했는지를 평점 등과 같이 아이템 만족 정도를 알 수 있는 데이터이다. 즉, 다중 클래스 피드백에는, 단일 클래스 피드백보다 사용자 선호도와 관련

된 더 많은 정보가 내포되어 있다. [12,13]

본 논문은 다중 클래스 환경에서 최신 지식 증류 기법 연구들 간의 추천 정확도를 비교하며, 어떤 방법이 다중 클래스 환경에서 더 효과적인지를 확인한다. 본 논문의 구성은 다음과 같다. 2 장에서는, 추천 시스템을 위해 제안된 기존의 최신 지식 증류 기법들에 대해 소개한다. 3 장에서는, 해당 방법들의 추천 정확도 개선 효과를 확인하기 위해 수행한 실험의 결과를 보인다. 마지막으로 4 장에서는, 본 논문의 결론 및 향후 연구에 대한 내용을 제시한다.

### 2. 추천 시스템을 위한 지식 증류 기법

추천 시스템을 위한 지식 증류 기법은 교사 모델로부터 증류되는 지식의 종류에 따라, 크게 예측 선호도 기반의 지식 증류 기법과 잠재적 특징 기반의 지식 증류 기법으로 구분될 수 있다.

#### 2.1 예측 선호도 기반의 지식 증류 기법

예측 선호도 기반의 지식 증류 기법은 학습된 교사 모델이 예측한 아이템에 대한 사용자 선호도를, 학생 모델에게 전달할 지식으로서 사용하는 방법이다. 학생 모델에게 전달될 아이템들 및 해당 아이템들에 대한 예측 선호도 값 결정 방법에 따라, RD, CD, RRD,

<sup>1</sup> 교신저자

BD [7-10]의 네 가지 연구들이 존재한다.

**Ranking Distillation (RD)** [7]은 처음 제안된 추천 시스템을 위한 지식 증류 기법으로, 사용자가 아직 이용하지 않은 아이템들 중, 교사 모델의 예측 선호도 기준 상위  $k$  개의 아이템들을 학생 모델에게 전달될 지식으로써 이용한다. 먼저, 학습 데이터를 통해 교사 모델이 각 사용자가 아직 이용하지 않은 아이템들에 대한 순위를 예측하도록 학습한다. 다음, 학생 모델은 학습 데이터와 학습된 교사 모델로부터 전달받은 아이템들을 활용하여 학습을 진행한다. 이 때, 전달받은  $k$  개의 아이템들은 사용자가 선호하는 아이템들일 것이라고 가정한다. 때문에, 단일 클래스 환경에서는 이들에게 ‘1’을 부여하여 학생 모델에게 전달하지만, 다중 클래스 환경에서는 ‘5’를 부여하여 학생 모델에게 전달한다.

**Collaborative Distillation (CD)** [8]은 먼저 RD의 한계로서, 교사 모델이 예측한 상위  $k$  개의 아이템들에 모두 같은 값을 부여하여 학생 모델에게 전달한다는 점을 지적한다. RD와 달리 CD는 교사 모델이 예측한 아이템들의 랭크를 기반으로 샘플링한  $k$  개의 아이템들(rank-aware sampling)에 대해 교사 모델이 예측한 선호도 값을 증류될 지식으로 사용한다. 그리고, 학생 모델은 학습 데이터와 교사 모델로부터 전달받은 아이템들 및 그에 대한 예측 선호도 값들을 통해 학습된다.

**Relaxed Ranking Distillation (RRD)** [9]은 사용자가 이용한 적 없는 아이템들에 대해, 학습된 교사 모델이 예측한 선호도를 바탕으로, 사용자의 관심/무관심 아이템들을 구분한다. 그리고, 이들에 대한 랭킹 정보를 학생 모델에게 지식으로서 전달한다. 학생 모델은 교사 모델로부터 전달받은 예측 랭킹 리스트를 바탕으로, 관심/무관심 아이템들을 잘 구분할 수 있는 방향으로 학습된다.

**Bidirectional Distillation (BD)** [10]은 사용자가 이용한 적 없는 아이템들에 대해, 각각 교사 모델과 학생 모델이 예측한 랭킹 정보를 지식으로서 구성한다. 이전의 다른 방법들과 달리 BD는 교사 모델과 학생 모델을 함께 학습시킨다. 즉, 각 모델이 같은 아이템에 대해 예측한 랭킹들 간의 차이(rank-discrepancy)를 통해, 상대 모델이 잘 예측한 아이템들에 대해 서로 잘 예측할 수 있도록 교사 모델과 학생 모델을 학습한다.

## 2.2 잠재적 특징 기반의 지식 증류 기법

잠재적 특징 기반의 지식 증류 기법은 교사 모델이 추론한 사용자/아이템에 대한 잠재적 특징들을 학생 모델에게 전달할 지식으로서 사용하는 방법이다. 잠재적 특징 추론 방법 및 학생 모델로의 전달 방법에

따라, DE [9], TD [11]의 두 가지 연구들이 존재한다.

**Distillation Experts (DE)** [9]는 교사 모델이 추론한 사용자/아이템 임베딩 벡터들을 통해 학생 모델에게 전달될 지식을 구성한다. DE는 교사 모델이 학습 데이터를 활용해 사용자/아이템의 임베딩 벡터들을 추론하도록 한다. 이 때, 학생 모델은 교사 모델의 임베딩 벡터와 학습데이터를 활용해 학생 모델의 잠재적 특징이 교사 모델의 잠재적 특징과 비슷해지도록 학습한다.

**Topology Distillation (TD)** [11]은 아이템들 간의 위상적(topological) 관계 정보로 증류될 지식을 구성한다. 이를 위해 TD에서의 교사 모델은 아이템들 간의 위상 구조를 활용하며, 학생 모델에 의해 추론된 아이템들 간의 위상적 관계가 교사 모델에 의해 구축된 위상 구조와 비슷해지는 방향으로 학생 모델을 학습한다.

우리는 다음 장에서, 기존 지식 증류 기법 방법들이 다중 클래스 환경에서 갖는 추천 정확도를 실험적으로 확인 및 비교하고자 한다.

## 3. 추천 정확도 평가

### 3.1 실험 환경

우리는 추천 시스템에서 보편적으로 사용되는 데이터셋들인, MovieLens 1M, 10M, Yelp를 통해 실험을 진행하였다. 데이터셋에 대한 통계적 정보는 표 1에 나온 바와 같다. 추천 시스템을 위한 협업필터링 모델로는 CDAE (Collaborative Denoising Auto-Encoder) [14]와 NeuMF (Neural Matrix Factorization) [15]를 사용했다. 모델 평가 프로토콜로는 5 겹 교차 검증방법을 사용했으며, 메트릭으로는 nDCG@10을 사용했다.

<표 1> 데이터셋에 대한 통계적 정보

	사용자 수	아이템 수	평점 수
ML 1M	6,041	3,953	1,000,209
ML 10M	69,878	10,677	10,000,054
Yelp	25,677	25,815	731,671

### 3.2 실험 결과

표 2에서 Teacher는 교사 모델을 의미하며, Student는 지식 증류 기법을 사용하지 않은 학생 모델과 크기만 같은 작은 모델을 의미한다. 기존의 지식 증류 기법들 중, DE-RRD는 예측 선호도 기반의 지식 증류 기법인 RRD와 잠재적 특징 기반의 지식 증류 기법인 DE가 함께 사용된 방법을 가리킨다 [10]. 표 2와 표 3을 통해서 우리는 다중 클래스 환경에서도 지식 증류 기법이 효과적으로 적용될 수 있음을 확인했다.

대부분의 경우 지식 증류 기법을 적용한 방법들이 Student 보다 CDAE 를 활용한 경우, 1.7%에서 136.2% 개선되는 것을 확인했다. NeuMF 를 사용한 경우에는 Student 보다 지식 증류 기법을 적용한 방법들이 6.3%에서 193.2% 개선되는 것을 확인했다. 표 2 에서는 전반적으로 RD 가 가장 높은 정확도를 보였고, TD 가 가장 낮은 정확도를 보였다. 반면, 표 3 에서는 CD 가 가장 높은 정확도를 보였고, RD 가 가장 낮은 정확도를 보였다.

<표 2> 추천 시스템을 위한 지식 증류 기법들 간의 추천 정확도 비교 (CDAE)

	ML 1M	ML 10M	Yelp
Teacher	0.1243	0.1543	0.0281
Student	0.0466	0.0534	0.0094
RD	<b>0.0698</b>	<b>0.1133</b>	0.0214
CD	0.0474	0.1119	0.0205
DE-RRD	0.0462	0.1067	0.0208
BD	0.0585	0.0969	<b>0.0222</b>
TD	0.0567	0.0989	0.0205

<표 3> 추천 시스템을 위한 지식 증류 기법들 간의 추천 정확도 비교 (NeuMF)

	ML 1M	ML 10M	Yelp
Teacher	0.1930	0.1361	0.0165
Student	0.1677	0.0413	0.0111
RD	0.1814	0.0876	0.0118
CD	<b>0.2070</b>	<b>0.1211</b>	0.0132
DE-RRD	0.2018	0.1137	<b>0.0135</b>
BD	0.1821	0.0784	0.0126
TD	0.2007	0.0899	0.0122

#### 4. 결론

본 논문에서는 추천 시스템을 위한 지식 증류 기법들을 다중 클래스 환경에 적용했을 때, 모두 효과적으로 동작함을 확인했다. 단일 클래스 환경에서 제일 효과적이었던 방법인 TD 가 다중 클래스 환경에서 가장 효과적이지는 않은 결과를 확인하고, 이후 다중 클래스 환경과 단일 클래스 환경의 차이를 고려하여 다중 클래스 환경에 적용될 수 있는 추천 시스템을 위한 지식 증류 기법들에 대한 연구도 진행해보고자 한다.

#### 사사

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구이고

(No.NRF-2020R1A2B5B03001960), 2018 년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No.2018R1A5A7059549). 또한, 2022 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2022-0-00352).

#### 참고문헌

- [1] S. Sedhain et al, "Autorec: Autoencoders meet collaborative filtering," in *International Conference on World Wide Web Companion*, 2015
- [2] D. Kim et al, "Convolutional matrix factorization for document context-aware recommendation," in *ACM Conference on Recommender Systems*, 2016
- [3] G. Hinton et al, "Distilling the knowledge in a neural network," in *CoRR*, abs/1503.02531, 2015.
- [4] J. Cui et al, "Knowledge distillation across ensembles of multilingual models for low-resource languages," in *IEE Conference on Acoustics, Speech and Signal Processing*, 2017
- [5] M.Hu et al, "Attention-guided answer distillation for machine reading comprehension," in *Conference on Empirical Methods in Natural Language Processing*, 2018
- [6] G. Aguilar et al, "Knowledge distillation from internal representations," in *AAAI Conference on Artificial Intelligence*, 2020
- [7] J. Tang and K. Wang, "Ranking distillation: Learning compact ranking models with high performance for recommender system," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2289-2298.
- [8] J. Lee et al, "Collaborative distillation for top-N recommendation," in *IEEE International Conference on Data Mining*, 2019, pp. 369-378.
- [9] S. Kang et al, "DE-RRD: A knowledge distillation framework for recommender system," in *ACM International Conference on Information and Knowledge Management*, 2020, pp.605-614
- [10] W. Kweon et al, "Bidirectional distillation for top-K recommender system," in *International World Wide Web Conference*, 2021, pp.3861-3871
- [11] S. Kang et al, "Topology distillation for recommender system," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2021, pp.829-839
- [12] Y. Hu et al, "Collaborative filtering for implicit feedback datasets," in *IEEE International Conference on Data Mining*, 2008
- [13] Y. Koren, "Factorization meets the neighborhood: a manufactured collaborative filtering model," in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2008
- [14] Y. Wu et al, "Collaborative denoising auto-encoders for top-N recommender systems," in *ACM International Conference on Web Search and Data Mining*, 2016, pp.153-162
- [15] X. He et al, "Neural collaborative filtering," in *International Conference on World Wide Web*, 2017, pp. 173-182