

1. GTN, LightGCN: 1000epochs 실험

1.1. 세팅

```
=====
810128 interactions for training
217242 interactions for testing
gowalla Sparsity : 0.0008396216228570436
gowalla is ready to go
=====CONFIG=====
{'A_n_fold': 100,
 'A_split': False,
 'K': 3,
 'args': Namespace(K=3, a_fold=100, alpha=0.3, alpha1=0.25, alpha2=0.25, avg=0, beta=0.5, bpr_batch=4096, comme
 'bigdata': False,
 'bpr_batch_size': 4096,
 'dataset': 'gowalla',
 'decay': 0.0001,
 'dropout': 0,
 'epochs': 1001,
 'keep_prob': 0.6,
 'lambda2': 4.0,
 'latent_dim_rec': 256,
 'lr': 0.001,
 'multicore': 1,
 'pretrain': 0,
 'test_u_batch_size': 100}
cores for test: 24
comment: gtn
tensorboard: 1
LOAD: 0
Weight path: /home/hwirc/Internship/GTN-SIGIR2022/checkpoints_GTN
Test Topks: [20]
using bpr loss
=====END=====
```

[Config1: GTN-1000epochs config]

```
loading [./data/gowalla]
=====
810128 interactions for training
217242 interactions for testing
gowalla Sparsity : 0.0008396216228570436
gowalla is ready to go
=====config=====
{'A_n_fold': 100,
 'A_split': False,
 'bigdata': False,
 'bpr_batch_size': 4096,
 'decay': 0.0001,
 'dropout': 0,
 'keep_prob': 0.6,
 'latent_dim_rec': 256,
 'lightGCN_n_layers': 3,
 'lr': 0.001,
 'multicore': 1,
 'pretrain': 0,
 'test_u_batch_size': 100}
cores for test: 24
comment: lgn
tensorboard: 1
LOAD: 0
Weight path: ./checkpoints
Test Topks: [20]
using bpr loss
=====end=====
Model: lgn
=====
```

[Config2: LightGCN-1000epochs config]

실험은 [Config1]과 [Config2]로 설정된 두 모델에 대해 진행됐다. batch size를 늘리는 것만으로도 실험 속도는 크게 향상됐다. 기존 실험, batch size를 512로 설정했을 때보다 4배 이상 빠른 속도로 실험이 진행됐다.

1.2. 실험 결과

```

618 + [TEST]
619 + =====
620 +
621 + Testing EPOCH[361/1001] | Results Top-k (pre, recall, ndcg): 0.057, 0.18719, 0.15777
622 +
623 + EPOCH[361/1001] loss 0.0125 0.0048 0.0077 - |Sample:9.09| | 01:44mins
624 + EPOCH[362/1001] loss 0.0125 0.0048 0.0077 - |Sample:8.73| | 01:17mins
625 + EPOCH[363/1001] loss 0.0126 0.0049 0.0077 - |Sample:9.16| | 01:18mins
626 + EPOCH[364/1001] loss 0.0125 0.0048 0.0077 - |Sample:8.74| | 01:17mins
627 + EPOCH[365/1001] loss 0.0125 0.0048 0.0077 - |Sample:8.97| | 01:17mins
628 + EPOCH[366/1001] loss 0.0125 0.0048 0.0077 - |Sample:8.78| | 01:19mins
629 + EPOCH[367/1001] loss 0.0126 0.0049 0.0077 - |Sample:8.98| | 01:19mins
630 + EPOCH[368/1001] loss 0.0125 0.0048 0.0077 - |Sample:8.80| | 01:19mins
631 + EPOCH[369/1001] loss 0.0125 0.0048 0.0077 - |Sample:9.00| | 01:19mins
632 + EPOCH[370/1001] loss 0.0126 0.0049 0.0077 - |Sample:8.75| | 01:19mins
633 + =====
634 + [TEST]
635 + =====
636 +
637 + Testing EPOCH[371/1001] | Results Top-k (pre, recall, ndcg): 0.05712, 0.18762, 0.15807
638 +
639 + EPOCH[371/1001] loss 0.0126 0.0049 0.0077 - |Sample:8.95| | 01:48mins
640 + EPOCH[372/1001] loss 0.0125 0.0048 0.0077 - |Sample:8.71| | 01:19mins

```

[결과1: GTN]

```

+ =====
+ [TEST]
+ =====
+ {'precision': array([0.05752395]), 'recall': array([0.18749521]), 'ndcg': array([0.15933124])}
+ EPOCH[361/1001] loss0.010-|Sample:8.94| | 01:19mins
+ EPOCH[362/1001] loss0.010-|Sample:8.76| | 00:56mins
+ EPOCH[363/1001] loss0.010-|Sample:8.78| | 00:56mins
+ EPOCH[364/1001] loss0.010-|Sample:8.89| | 00:51mins
+ EPOCH[365/1001] loss0.010-|Sample:8.61| | 00:56mins
+ EPOCH[366/1001] loss0.010-|Sample:8.73| | 00:56mins
+ EPOCH[367/1001] loss0.010-|Sample:8.79| | 00:51mins
+ EPOCH[368/1001] loss0.010-|Sample:8.87| | 00:56mins
+ EPOCH[369/1001] loss0.010-|Sample:8.71| | 00:56mins
+ EPOCH[370/1001] loss0.010-|Sample:8.83| | 00:55mins
+ =====
+ [TEST]
+ =====
+ {'precision': array([0.05770313]), 'recall': array([0.18795718]), 'ndcg': array([0.15978268])}
+ EPOCH[371/1001] loss0.010-|Sample:8.94| | 00:58mins
+ EPOCH[372/1001] loss0.010-|Sample:8.90| | 00:44mins
+ EPOCH[373/1001] loss0.010-|Sample:8.79| | 00:56mins
+ EPOCH[374/1001] loss0.010-|Sample:8.77| | 00:56mins
+ EPOCH[375/1001] loss0.010-|Sample:8.57| | 00:51mins
+ EPOCH[376/1001] loss0.010-|Sample:8.68| | 00:56mins
+ EPOCH[377/1001] loss0.010-|Sample:9.36| | 00:57mins
+ EPOCH[378/1001] loss0.010-|Sample:8.99| | 00:55mins
+ EPOCH[379/1001] loss0.010-|Sample:8.84| | 00:56mins
+ EPOCH[380/1001] loss0.010-|Sample:8.95| | 00:56mins
+ =====

```

[결과2: LightGCN-370epochs]

실험의 결과는 [결과 1], [결과 2]와 같다. GTN 의 경우 도중에 실험이 끊겼기 때문에 동일한 epochs 에서 비교를 진행했다. 실험 결과 표는 다음과 같다.

모델	370epochs	1000epochs
GTN	15.807%	-
LightGCN	15.978%	16.224%

[표 1]

1.3. 고찰

[표 1]과 같이 이번에도 LightGCN 은 GTN 보다 우수한 성능을 보였다. 하지만 의문점은 LightGCN 의 성능이 논문에 제시된 수치보다도 높게 측정되었다는 점이다.

Table 3: Performance comparison between NGCF and LightGCN at different layers.

Dataset		Gowalla		Yelp2018		Amazon-Book	
Layer #	Method	recall	ndcg	recall	ndcg	recall	ndcg
1 Layer	NGCF	0.1556	0.1315	0.0543	0.0442	0.0313	0.0241
	LightGCN	0.1755(+12.79%)	0.1492(+13.46%)	0.0631(+16.20%)	0.0515(+16.51%)	0.0384(+22.68%)	0.0298(+23.65%)
2 Layers	NGCF	0.1547	0.1307	0.0566	0.0465	0.0330	0.0254
	LightGCN	0.1777(+14.84%)	0.1524(+16.60%)	0.0622(+9.89%)	0.0504(+8.38%)	0.0411(+24.54%)	0.0315(+24.02%)
3 Layers	NGCF	0.1569	0.1327	0.0579	0.0477	0.0337	0.0261
	LightGCN	0.1823(+16.19%)	0.1555(+17.18%)	0.0639(+10.38%)	0.0525(+10.06%)	0.0410(+21.66%)	0.0318(+21.84%)
4 Layers	NGCF	0.1570	0.1327	0.0566	0.0461	0.0344	0.0263
	LightGCN	0.1830(+16.56%)	0.1550(+16.80%)	0.0649(+14.58%)	0.0530(+15.02%)	0.0406(+17.92%)	0.0313(+18.92%)

*The scores of NGCF on Gowalla and Amazon-Book are directly copied from Table 3 of the NGCF paper (<https://arxiv.org/abs/1905.08108>)

[표 2]

Table 2: The comparison of overall performance.

Datasets		Gowalla		Yelp2018		Amazon-Book		LastFM	
Metrics		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
Method	MF	0.1299	0.111	0.0436	0.0353	0.0252	0.0198	0.0725	0.0614
	NeuCF	0.1406	0.1211	0.045	0.0364	0.0259	0.0202	0.0723	0.0637
	GC-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224	0.0804	0.0736
	NGCF	0.156	0.1324	0.0581	0.0475	0.0338	0.0266	0.0774	0.0693
	Multi-VAE	0.1641	0.1335	0.0584	0.045	0.0407	0.0315	0.078	0.07
	DGCF	0.1794	0.1521	0.064	0.0522	0.0399	0.0308	0.0794	0.0748
	LightGCN	0.1823	0.1553	0.0649	0.0525	0.042	0.0327	0.085	0.076
	GTN (Ours)	0.187	0.1588	0.0679	0.0554	0.045	0.0346	0.0932	0.0857
Relative Improvement (%)		2.59	2.26	4.62	5.59	7.15	5.95	9.61	12.77

[표 3]

[표 2]와 [표 3]은 각각 LightGCN, GTN 논문이 밝힌 각 모델들의 성능이다. 주목할 점은 LightGCN 의 성능(ndcg)값이 **15.55%** 내외였다는 점이다.

실험 결과와 비교할 때, LightGCN 의 경우 370 epochs 만으로도 논문에서 밝힌 ndcg 값을 넘는 **15.98%**를 기록했고 GTN 의 경우 논문에서 밝힌 값보다는 작은 **15.81%**를 기록했다.

이번 실험으로 GTN 모델에 대한 논문 수준의 재현은 가능하다고 판단했다. 그리고 LightGCN 의 수치에 대한 관찰을 추후 과제로 설정했다.

2. LightGCN: Noise 실험

2.1. 실험 세팅과 결과

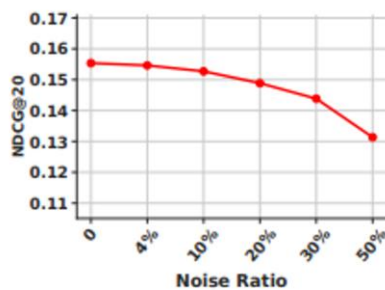
```
# =====  
random_item=[]  
for _ in range(10):  
    random_item.append(random.randint(0, 40980))  
items.extend(random_item)  
# =====
```

[코드1: noise 추가 코드]

noise 추가 환경을 만들기 위해 [코드1]과 같이 각 유저별 **10개씩의 랜덤 아이템을 추가했다**. 이때 noise를 추가한 데이터셋을 train에만 국한한 경우, train, test 모두에 추가한 경우로 나눠 실험 결과를 확인했다. 그 결과는 아래 [표4]와 같다.

데이터셋	Noise 추가 비율	성능	성능 감소 정도
기존 데이터셋(gowalla)	-	12.25%	-
Train 데이터셋에만 추가	37%	11.36%	8%
Train, Test 데이터셋에 모두 추가	58%	6.732%	48%

[표4]



(c) Gowalla-NDCG@20

[그래프1]

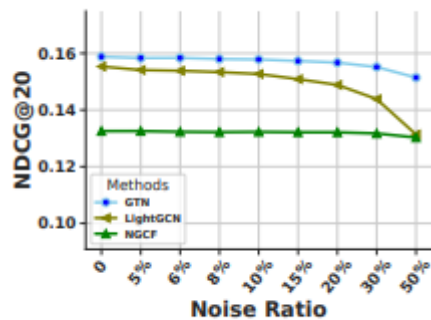
[그래프1]은 GTN 논문에서 밝힌 LightGCN의 noise 증가 비율에 따른 성능 감소 정도 결과이다. [그래프1]과 비교할 때 train 데이터셋에 noise를 추가했을 때의 성능 감소 정도가 비슷한 것을 확인했다. 따라서 GTN 논문의 성능 감소 실험을 재현했다고 판단했다.

2.2. 고찰

GTN 논문을 다시 읽던 중 noise 실험은 test 데이터셋에 random한 item을 추가하는 것임을 알게 됐다. 그리고 이에 대한 결과를 [그래프2]에서 밝혔다. 이에 대한 재현 실험이 필요할 것이라고 생각했다.

we perturb the test data by randomly injecting a certain ratio of edges and evaluate the recommendation performance of the models trained on clean data. In addition, we vary the ratio of noisy edges

[인용1]



(e) Gowalla - NDCG@20

[그래프2]