Music Recommendation using Graph Convolutional Networks

Final Postgraduate Project on IA and Deep Learning

[Artificial Intelligence with Deep Learning Postgraduate (UPC)]

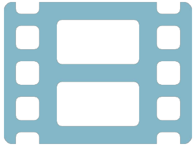
Authors: Miguel Ibero, Rafael Pérez, Abel Martinez
Advisor: Paula Gómez

code repository https://github.com/miguelibero/aidl-nnrecomend

Structure Of The Presentation

1. What have we done?
2. Structure of datasets & evaluation metrics
3. Implemented Models
4. Reproducing *“Deep Collaborative Filtering...”* paper results
5. Incorporating Graph Convolutional Networks
6. Experiments with novel data
7. Addressing the cold start problem
8. Conclusions

1. What Have We Done?



Movielens 100k dataset

test & improve our models
compare with published results

Spotify Skip Challenge dataset

analyze and extract second dataset
train & obtain new results

2. Structure of Datasets & Evaluation Metrics

Interaction Dataset Format

	User	Item	Context 1	Context 2	Rating
	0	450	800	1201	5
	0	720	851	1201	4
	0	640	1121	1206	3
	1	402	800	1204	0
	1	788	803	1202	1
	2	590	800	1201	2
	...	...		...	...
min value	0	400	800	1201	0
max value	399	799	1200	1206	5
range	399	399	400	5	

Generating The Testset

User	Item	Context 1	Context 2	Rating
0	450	800	1201	5
0	720	851	1201	4
0	640	1121	1206	3
1	402	800	1204	0
1	788	803	1202	1
2	590	800	1201	2

trainset

User	Item	Context 1	Context 2	Rating
0	450	800	1201	5
0	720	851	1201	4
1	402	800	1204	0
2	590	800	1201	2

testset

User	Item	Context 1	Context 2	Rating
0	640	851	1206	3
1	788	803	1202	1

Trainset Negative Sampling

	User	Item	Context 1	Context 2	Rating
	0	450	800	1201	1
	0	580	800	1201	0
negative samples	0	721	800	1201	0
	0	590	800	1201	0
	0	720	591	1201	1
	0	560	591	1201	0
negative samples	0	421	591	1201	0
	0	537	591	1201	0

Evaluation Metrics

Hit Ratio (HR)

Measures whether the real test item is in the top positions of the recommendation list

Normalized Discounted Cumulative Gain (NDCG)

Measures the ranking quality which gives information about where in the raking is our real test item.

Coverage (COV)

Measures the amount of total items in the topk positions.

User	Item	Context 1	Context 2	Predicted
0	640	851	1206	3,2
0	580	800	1201	1,1
0	721	800	1201	2,1
0	590	800	1201	0,7
0	580	800	1201	0,2
0	721	800	1201	1,5
0	590	800	1201	2,7

take topk

when testing, the batch contains 1 real user item and the rest is negative samples of the same user

3. Implemented Models

Baseline

simple algorithm that recommends the most popular items to everyone

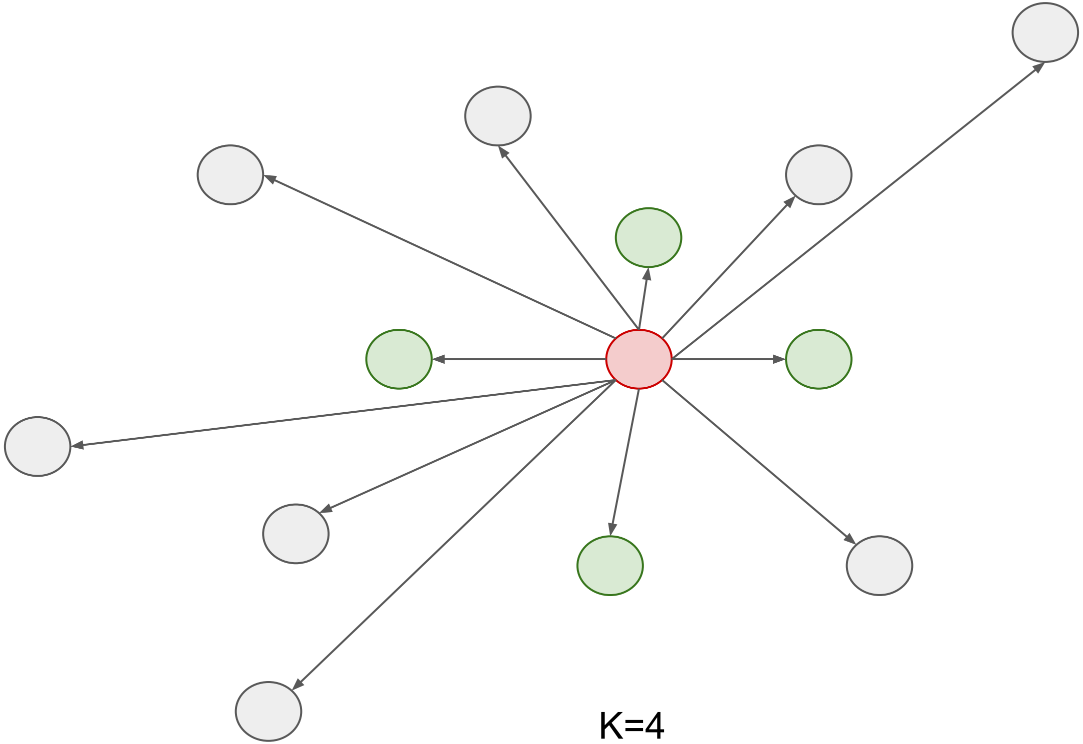
$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i$$

average item popularity

user bias

item bias

K Nearest neighbors



K=4

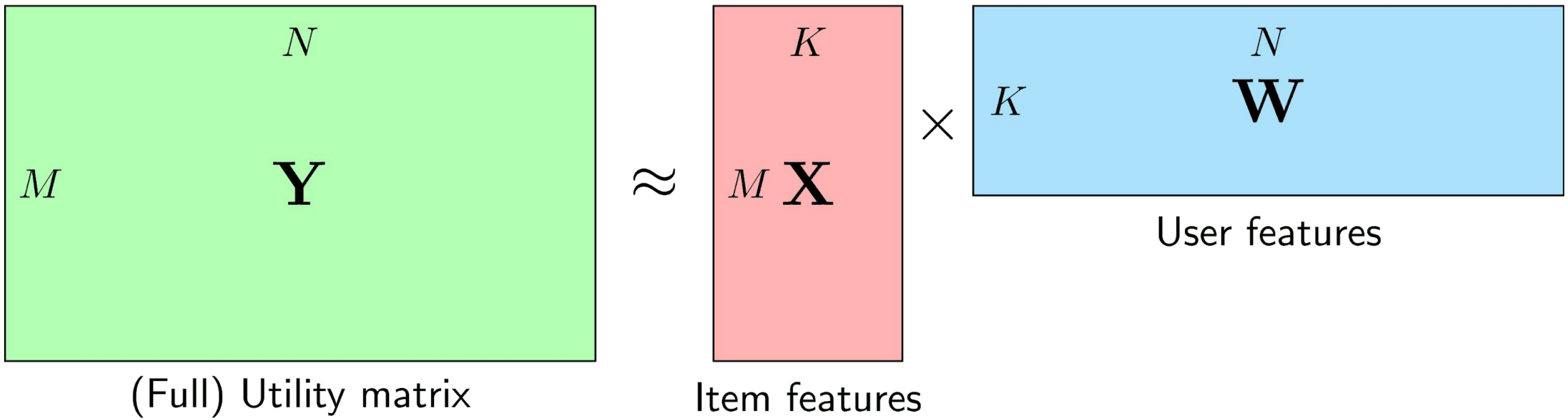
look for the most similar users based on their recommendations

$$\hat{r}_{ui} = \frac{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum\limits_{v \in N_i^k(u)} \text{sim}(u, v)}$$

see this paper

Matrix Factorization

$$\mathbf{Y} \approx \hat{\mathbf{Y}} = \mathbf{X}\mathbf{W}$$



Factorization Machine

$$\hat{y}(\mathbf{x}) := \boxed{w_0 + \sum_{i=1}^{n} w_i\, x_i} + \boxed{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle\, x_i\, x_j}$$

reformulation of the second part

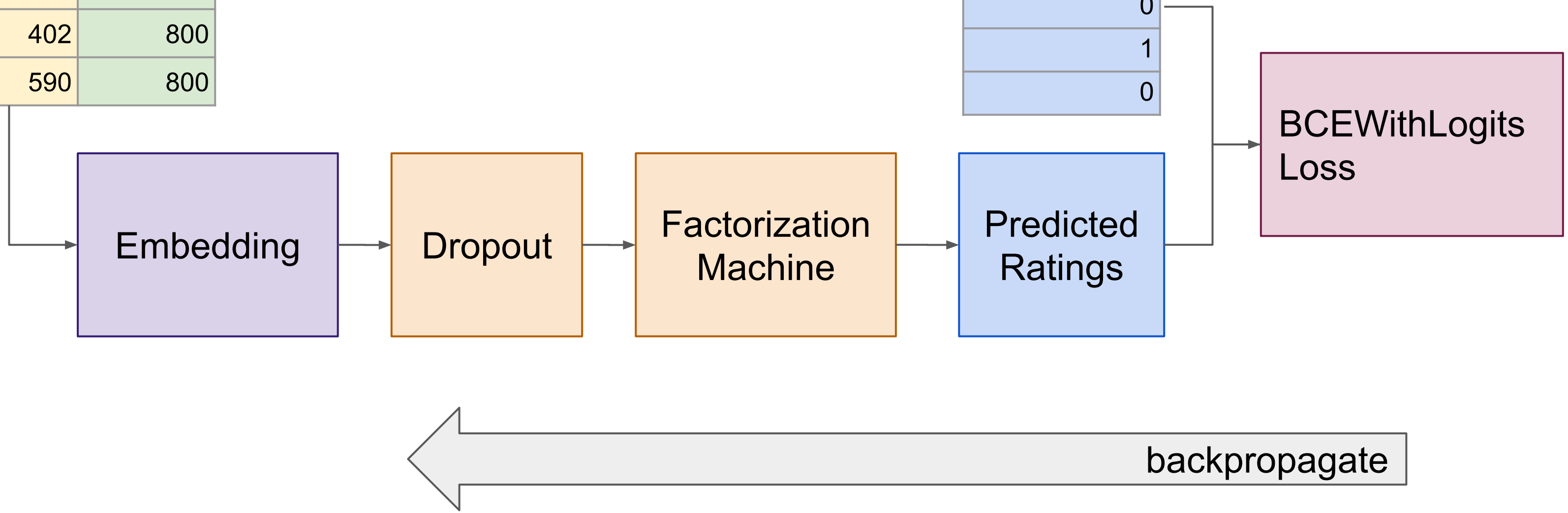
$$= \frac{1}{2} \sum_{f=1}^{k} \left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i \right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2 \right)$$

proposed in this 2010 paper

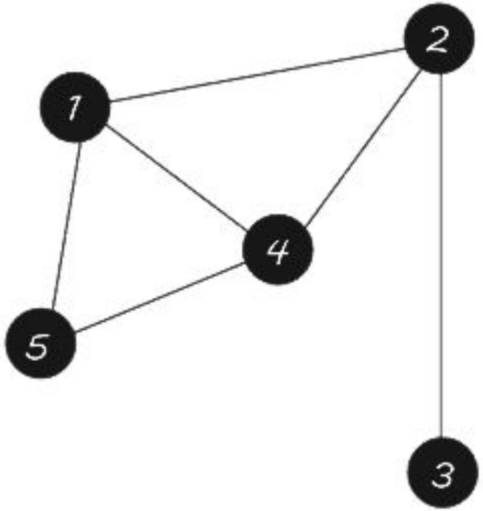
Factorization Machine Implementation

User	Item	Context 1
0	450	800
0	720	851
1	402	800
2	590	800

Rating
1
0
1
0



Graph Convolutional Networks



$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 & 1 \\ 5 & 1 & 0 & 0 & 1 & 0 \end{array}$$

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

presented in this paper

4. Reproducing “Deep Collaborative Filtering” paper results

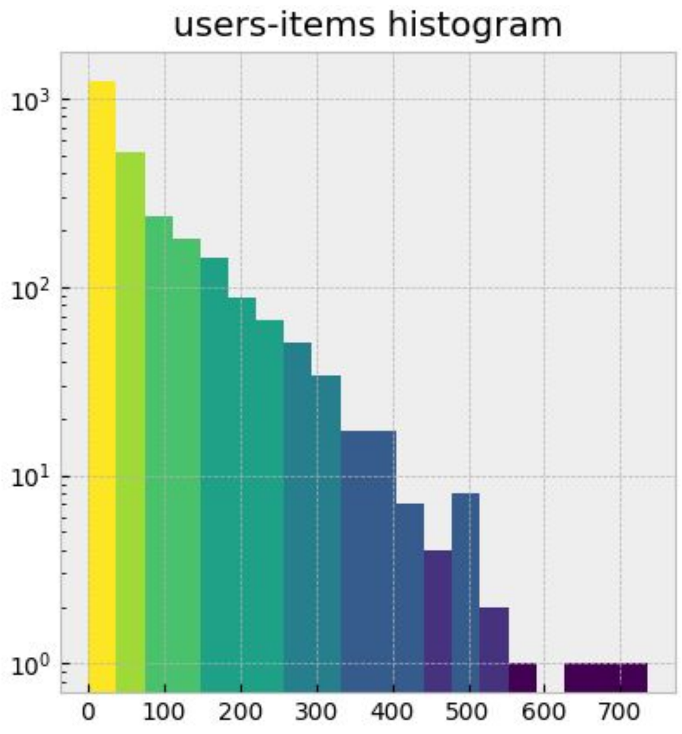
Approach for reproducing the paper

We found this 2019 paper that evaluates multiple recommender systems using the **Movielens 100k** dataset:

- 100,000 interactions (ratings 1-5)
- 943 users and 1682 movies
- each user has rated at least 20 movies

to compare the paper results with our models we maintain the same parameters

- 64 hidden dimensions
- 10 trainset negative samples
- 99 testset negative samples
- 10 topk



dataset can be found here

Bayesian Personalized Ranking Loss (BPR)

instead of comparing each rating prediction to the expected one,
we create positive-negative pairs and try to separate the predictions

```
pred_pos = model(pos_interactions)
pred_neg = model(neg_interactions)
loss = -(pos-neg).sigmoid().log().mean()
```

positive interaction (repeated) | negative interactions

User	Item	Context 1	Context 2	User	Item	Context 1	Context 2
0	450	800	1201	0	580	800	1201
0	450	800	1201	0	721	800	1201
0	450	800	1201	0	590	800	1201

presented in [this 2009 paper]

Paper Results Comparison

	ItemPop	baseline	ItemKNN	knn	FMG	FM	NeuACF	NeuACF++
type	paper	ours	paper	ours	paper	ours	paper	paper
HR@10	0,3998	0,4051	0,5891	0,5716	0,6373	0,6458	0,6846	0,6915
NDCG@10	0,2264	0,2191	0,3283	0,3422	0,3588	0,3658	0,4068	0,4092

we reproduce very similar results to the paper

5. Incorporating Graph Convolutional Networks

Factorization Machine + Graph Convolutional Network

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \boxed{\langle \mathbf{v}_i, \mathbf{v}_j \rangle} x_i x_j$$

$v_i = GCN_i(matrix, x_i)$

we implemented it using two types:

```
torch_geometric.nn.GCNConv
torch_geometric.nn.GATConv
```

	User_0 .... User_Nu	Item_0 ... Item_Ni	Context_0 ... Context_Nc
User_0 ... User_Nu	0	Interactions user-item	Interactions user-context
Item_0 ... Item_Ni	Interactions user-item	0	Interactions item-context
Context_0 ... Context_Nc	Interactions user-context	Interactions item-context	0

proposed in [this paper]

Adding Previous Item As Context

	User	Item	Previous Item
	0	450	-1
	0	720	450
	0	640	720
			640
min value	0	400	
max value	399	799	
range	399	399	

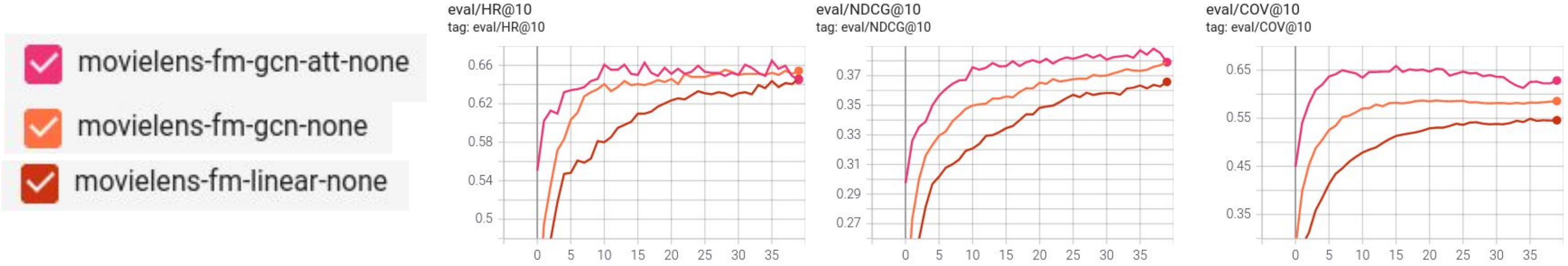
Previous Item
-1+1+800
450-400+1+800
720-400+1+800

800
1200
400

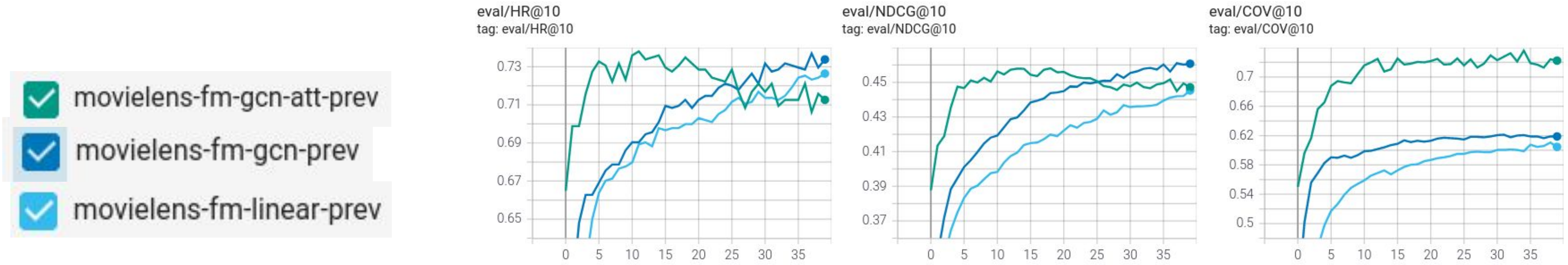
Previous Item
800
851
1121

Evaluation Graphs

without context



with previous item context



Final Results Comparison

adding GCN and GCN+attention we improve

	FMG	FM	FM-GCN	FM-GCN-ATT	NeuACF	NeuACF++
type	paper	ours	ours	ours	paper	paper
HR@10	0,6373	0,6458	0,6543	0,6596	0,6846	0,6915
NDCG@10	0,3588	0,3658	0,3792	0,3883	0,4068	0,4092
COV@10		0,5458	0,5856	0,6225		

adding previous item as context we get even better results

	FM	FM-GCN	FM-GCN-ATT
type	ours	ours	ours
HR@10	0,7264	0,7370	0,7349
NDCG@10	0,4453	0,4611	0,4581
COV@10	0,6046	0,6165	0,7206

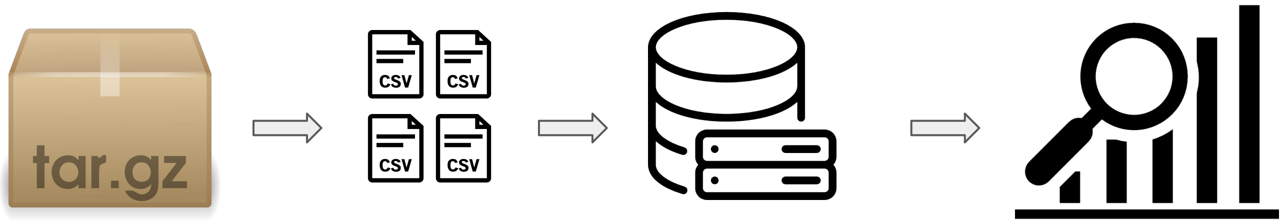
6. Experiments with novel data

Choosing a dataset

- Complete Training Set (56 Gb.)
- Minimally sized version of training set (17.2 Mb)

You can find this dataset here

Spotify DataSet Extraction



56 Gb.

256 Gb.
300 Files

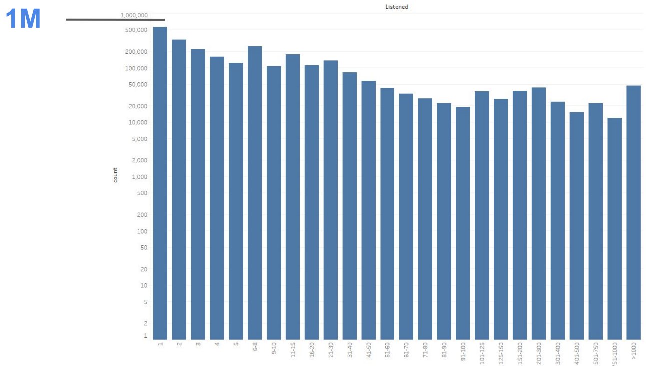
1000M Records
Track + Session
50M Sessions x 20 Songs
2.7M Diferent Songs

Analyze data

Will spotify mini dataset have a similar distribution?

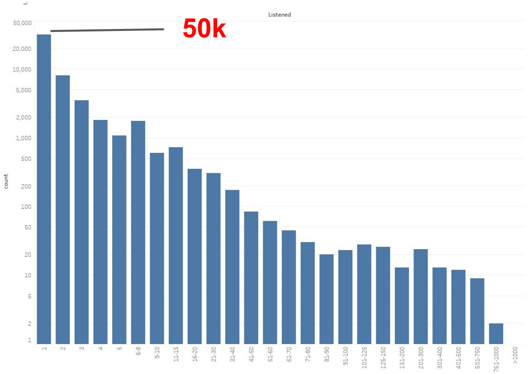
Lets check it.

Mini Data Set Construction



50M Sessions
2.7M Songs

Histogram for complete dataset

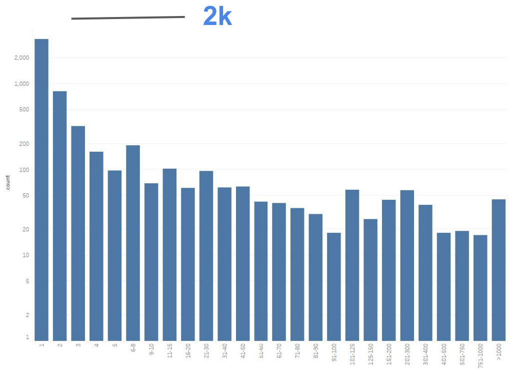


RANDOM

10k Random sessions
50k Songs

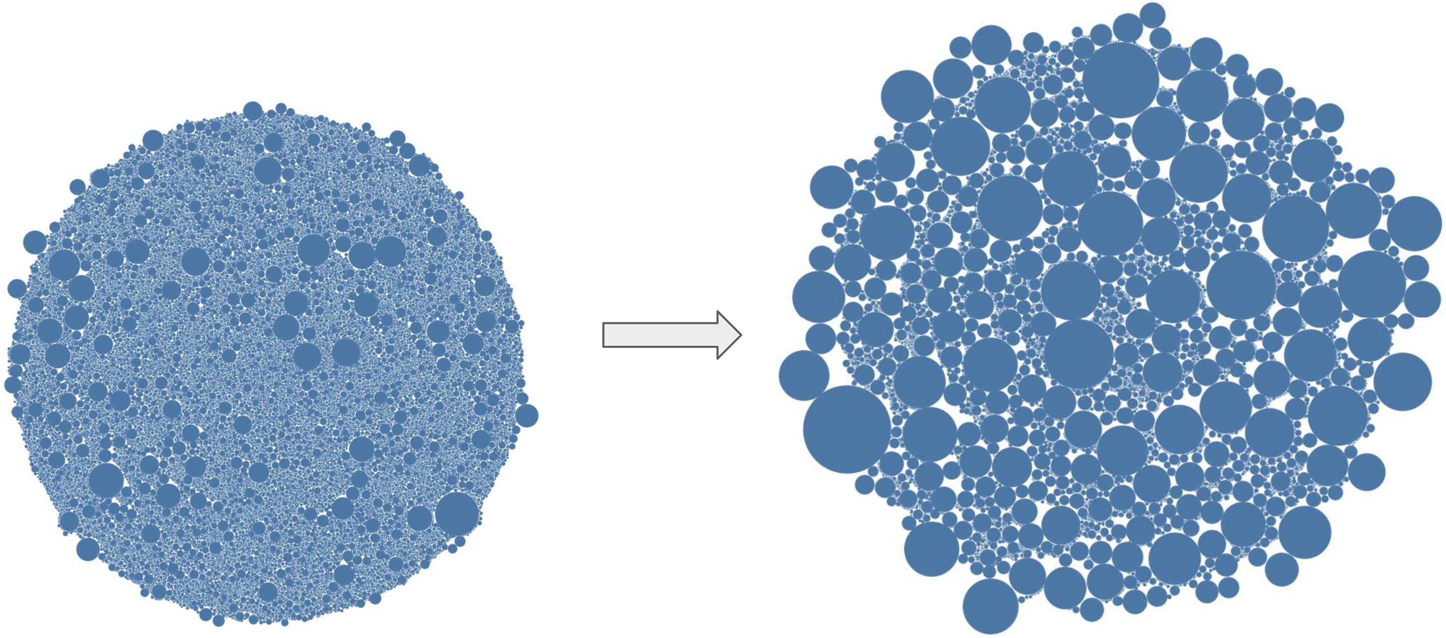
10k Random sessions
5.7k Songs

10k Random sessions
X Songs ?



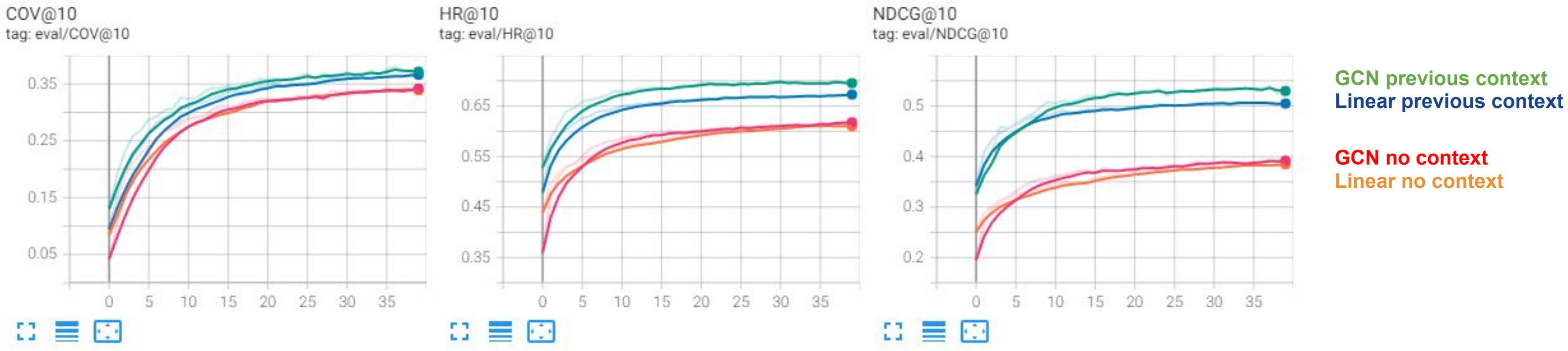
BALANCED

Mini Dataset Compare



RANDOM BALANCED

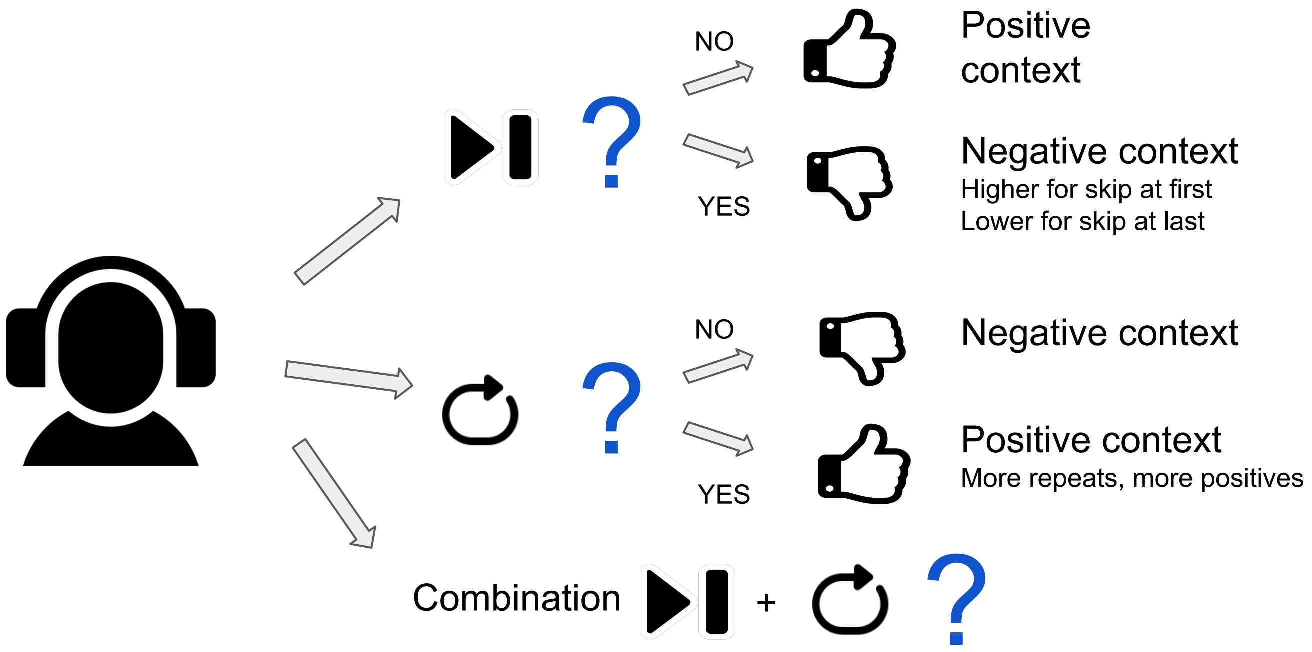
Spotify Results



- Results are very good, we are using all negatives available for test (2232 negatives average)
- Using previous item is significantly better than no context
- Using GCN is better than Linear
- GCN requires more computation than linear

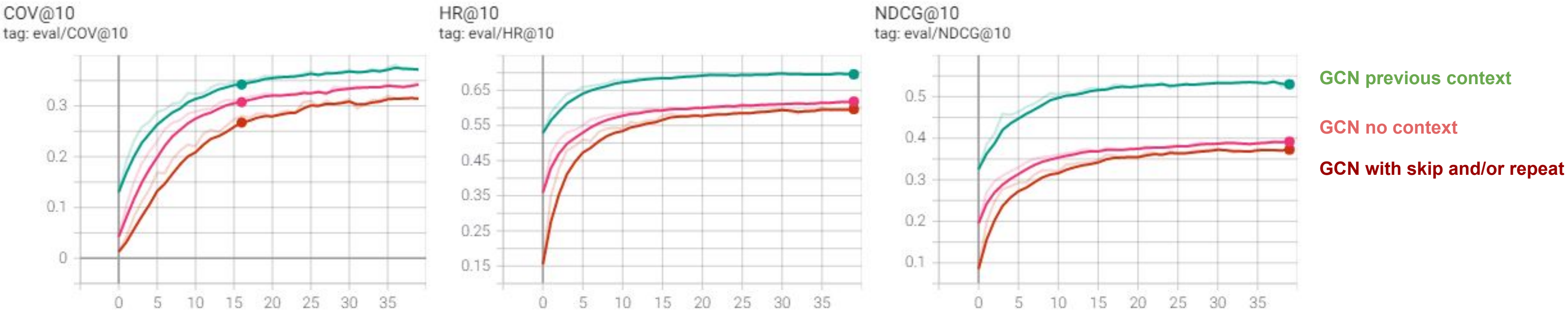
Searching context for Spotify sessions

Skip information
Repeated songs
Hour of day
...



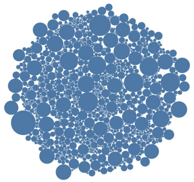
These context will improve user-item results ? We think it won’t improve results...

Using Skip as context...

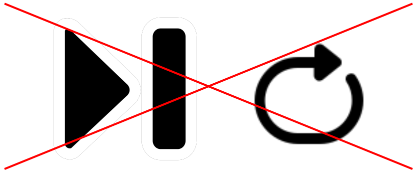


So, as we predict, skip or repeated songs not improves the results

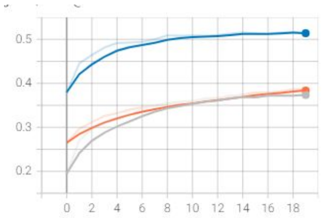
Conclusions working with Spotify Mini Dataset



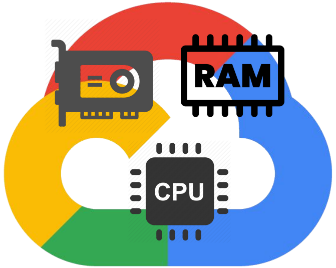
Mini artificial data set
Very good results
(Most probably better than full)



Skip or repeat is not good as context



GCN is the
best configuration

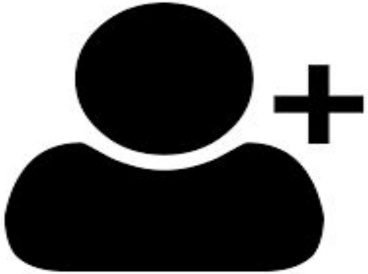


GCN requires a lot of compute requirements

GCN + Attention requires much more

7. Addressing the cold start problem

Cold Start Problem



Generating New Item Recommender Dataset

	~~User~~	Item	Previous
	~~0~~	450	800
	~~0~~	720	851
	~~0~~	640	1121
	~~1~~	402	800
	~~1~~	788	803
	~~2~~	590	800
min value	~~0~~	400	800
max value	~~399~~	799	1200
range	~~399~~	399	400

	Item	Previous
	~~50~~	~~400~~
	320	451
	240	721
	~~2~~	~~400~~
	388	403
	~~190~~	~~400~~
min value	0	400
max value	399	800
range	399	400

	Item	Previous
	320	450
	240	720
	388	402
min value	0	400
max value	399	799
range	399	399

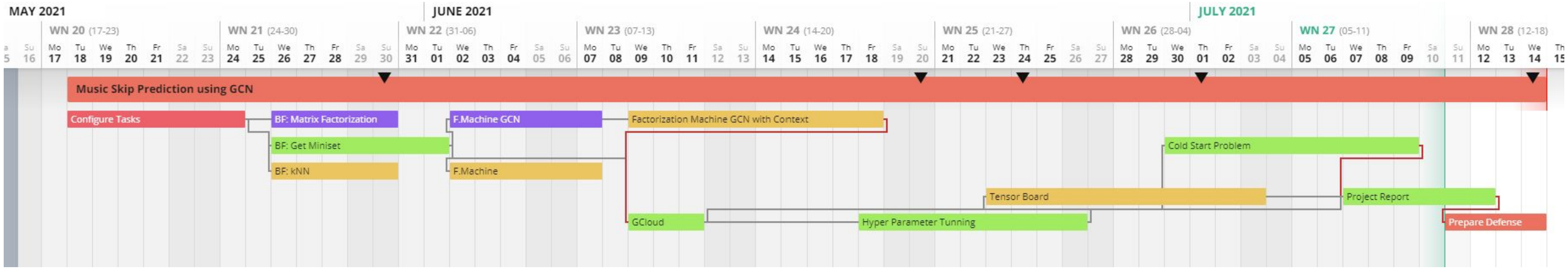
	Previous	Item
	50	720
	320	640
	2	788
min value	0	400
max value	399	799
range	399	399

New User Recommender

```
[mibero@snowden aidl-nnrecommend]$ nnrecommend recommend models/movielens_recommend.pth --label "star wars"
2021-07-11 22:28:30 reading model file...
2021-07-11 22:28:34 loaded idrange [1682 3364]
2021-07-11 22:28:34 loaded model of type <class 'nnrecommend.model.FactorizationMachine'>
2021-07-11 22:28:34 loaded 1682 items
2021-07-11 22:28:35 found item 49:
title                                                   Star Wars (1977)
release_date                                                 01-Jan-1977
link              http://us.imdb.com/M/title-exact?Star%20Wars%20(1977)
original_item_id                                                      50
2021-07-11 22:28:35 >>>
2021-07-11 22:28:35 looking for recommendations...
title                                                 Fargo (1996)
release_date                                           14-Feb-1997
link              http://us.imdb.com/M/title-exact?Fargo%20(1996)
original_item_id                                               100
title                                                        Return of the Jedi (1983)
release_date                                                               14-Mar-1997
link              http://us.imdb.com/M/title-exact?Return%20of%20the%20Jedi%20(1983)
original_item_id                                                                   181
title                                                   Godfather, The (1972)
release_date                                                      01-Jan-1972
link              http://us.imdb.com/M/title-exact?Godfather,%20The%20(1972)
original_item_id                                                          127
```

TOP 3 RECOMMENDATIONS

Conclusions



- we implemented the different recommender system models
- we reproduced paper model metrics for the movilens dataset
- we implemented BPRLoss, GCN and previous item context
- we applied the same models to the spotify dataset
- we showed that GCN embeddings improve the evaluation metrics
- we proposed solution to cold start problem

- It took much more time to tune than to implement
- We could not find a good additional context in the spotify dataset
- it would be interesting to implement Neural FM

Thanks!
Questions?