Recommender System

Davide Roznowicz

Goals of the project

Goals of the project

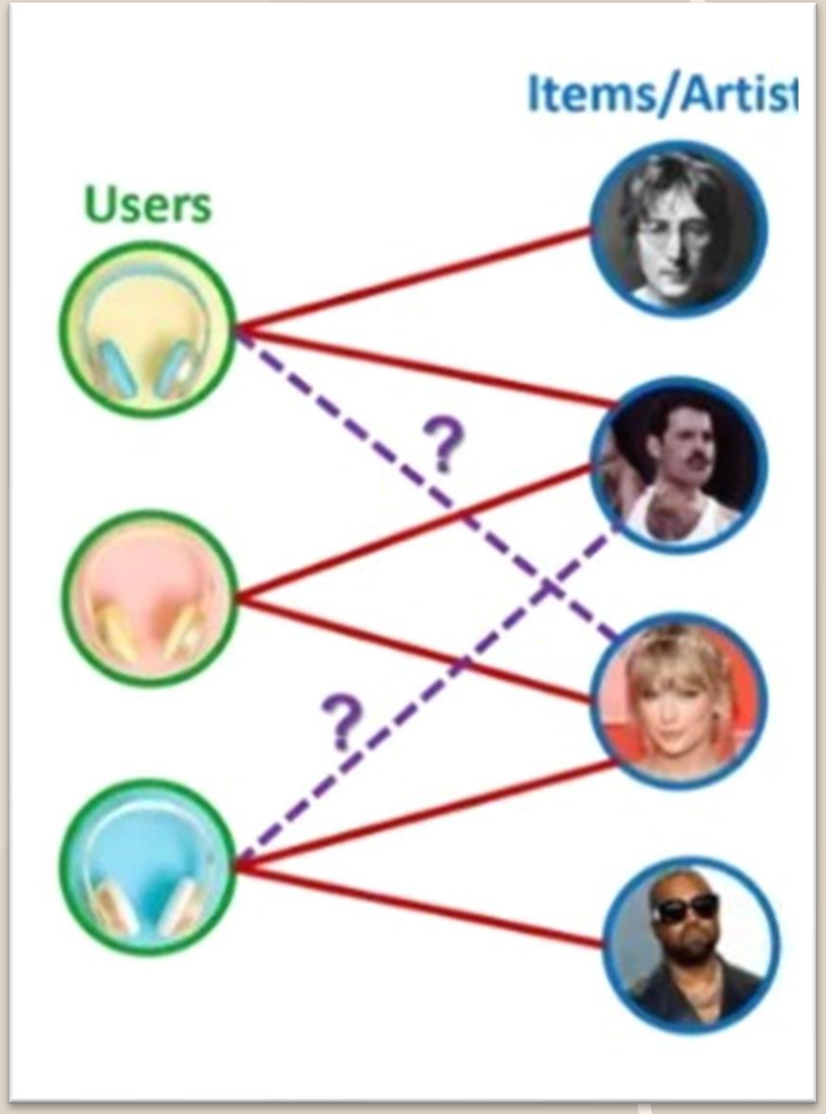
Build a Recommender System by following the project requirements:

1. Find a user-item dataset detailing which items the user has liked
2. Build a Weighted MF to find the user and item embedding
3. Allow to choose a user we should recommend items to
4. Return a ranking of documents

Focus

The system must incorporate **collaborative filtering:** recommending an item to the user is based on the extraction and exploitation of similarities among users and items via:

- ➢ item embedding
- ➢ user embedding



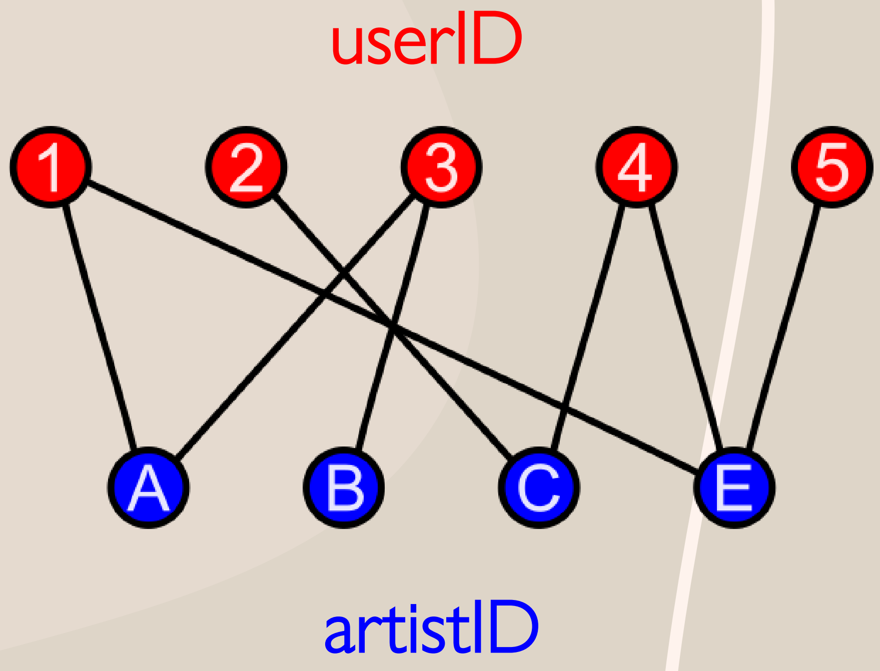
1. Dataset

Chosen Dataset

- Dataset extracted from the website https://www.last.fm, which is a mix of a radio and social networking platform
- It gathers information about users (**userID**) following/listening to some artists (**artistID**)
- If a user listens to an artist, the interaction matrix displays 1; otherwise 0

Structure of the dataset

- The dataset can be thought as a **bipartite graph**
- The users have no direct relation among each other; the same is true for the artists as they have no direct edge.
- user-artist connections create a **collaborative environment** in which similarities of interests can be extracted



Data pre-processing

While pre-processing the dataset, we realize that:

- it is better to downsize the dataset for a faster training

- there is always at least a user following a specific artist

userID	artistID
0	1
0	2
0	3
0	4
0	5
...	...
1812	83
1812	89
1812	99
1812	111
1812	122

id	name
0	Marilyn Manson
1	Duran Duran
2	Kylie Minogue
3	Daft Punk
4	New Order
...	...
121	MGMT
122	Led Zeppelin
123	Ramones
124	Avenged Sevenfold
125	Adele

2. Weighted Matrix Factorization via WALS

Problem statement

$$C \approx U \cdot V^T$$

$$\min_U \sum_{j=0}^{n-1} w_{ij}(c_{ij} - u_i^T v_j)^2 + \lambda \sum_{z=0}^{k-1} u_{iz}^2$$

We want to compute the embeddings U and V via minimization of the loss function

- C $(m \times n)$ is the interaction matrix (1 if the user liked the artist)
- U $(m \times k)$ is the user embedding (each row represents a user)
- V $(n \times k)$ is the item embedding (each row represents an artist)

WALS Algorithm

this algorithm can be easily parallelized:

Iterate over steps:

Iterate over j :

$$\hat{V}_j = (U^T W_j U + \lambda I)^{-1} U^T W_j C_j$$

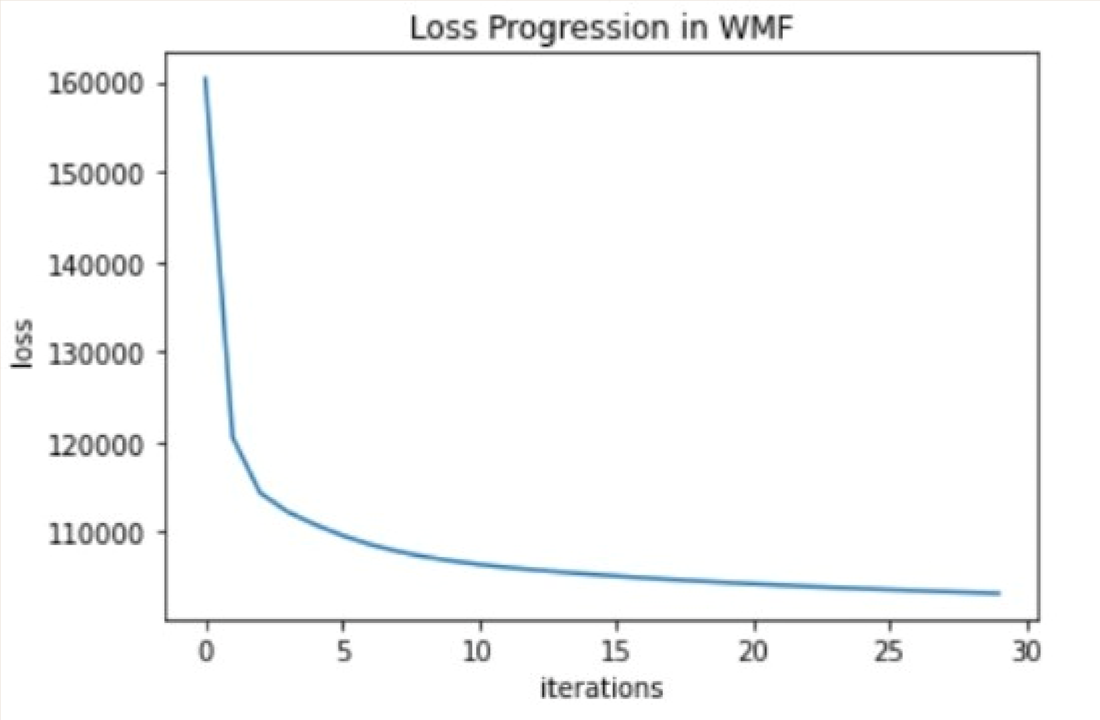
Iterate over i :

$$\hat{U}_i = (V^T W_i V + \lambda I)^{-1} V^T W_i C_i$$

fix V and update U

fix U and update V

Model Training and Convergence



We train the model and monitor the loss: after convergence, we can start making predictions

Explaining the weights

The higher the confidence weight and the user-perceived item similarity, the more influence that item has on "item_to_explain"

Predicted Score for User u on Item i: 0.7559506106122453

Item No.	User-Perceived Item Similairty	Confidence Weight	C
1	0.00201	1.0	0.0
2	-0.00034	41.0	1.0
3	0.00505	41.0	1.0
4	0.00135	41.0	1.0
5	-0.00099	41.0	1.0
6	0.0027	41.0	1.0
7	0.00474	41.0	1.0
8	0.00045	41.0	1.0
9	-0.00039	41.0	1.0
10	0.00078	41.0	1.0
11	0.00507	41.0	1.0
12	-6e-05	1.0	0.0
13	0.00428	1.0	0.0
14	-2e-05	1.0	0.0

3. Input a user

Input a user

- Select a user among the ones in the dataset
- Each user corresponds to a row of the matrix U

4. Ranking artists for the selected user

Ranking algorithm

- ❖ The prediction matrix is: $C_{WMF} = U \cdot V^T$
- ❖ The prediction for the i-th user is: $C_{WMF_i} = U_i \cdot V^T$
- ❖ Recommending an artist to a user is just a matter of sorting the scores of the i-th predicted row of C

artistName	score	top_ranked
Björk	1.082708	1
Crystal Castles	1.025994	2
Portishead	0.979306	3
Depeche Mode	0.969119	4
Massive Attack	0.966116	5
Yeah Yeah Yeahs	0.962088	6
Arcade Fire	0.942400	7
Sigur Rós	0.935863	8
MGMT	0.933362	9
Keane	0.931965	10

Cold Start

- **cold start for user**: among all the recommended artists, we might recommend for a half the freshest/newest and for the second half the best rated artists among all users.

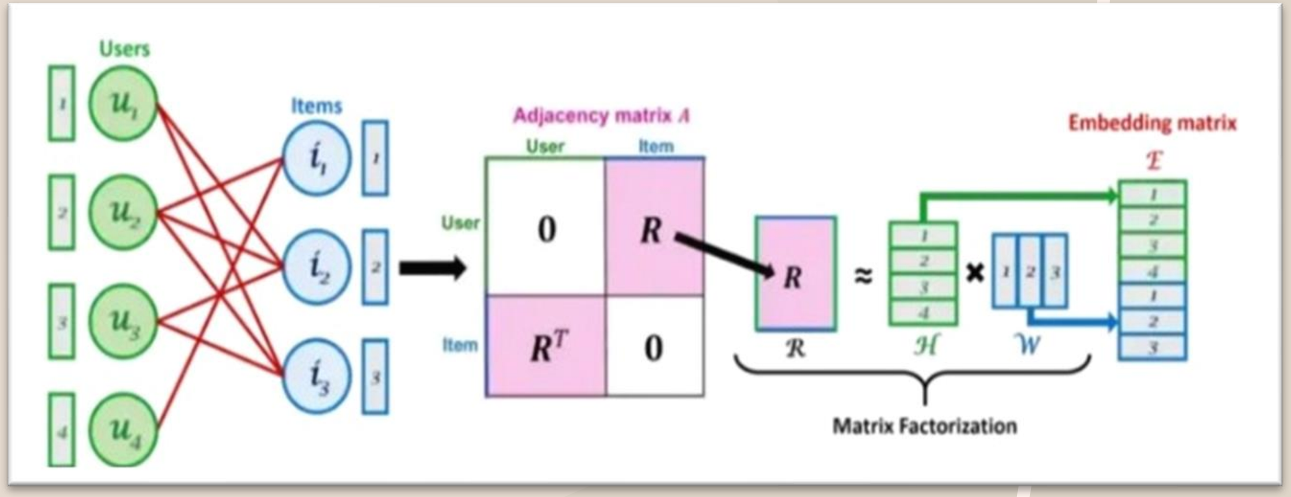
- **cold start for item**: recommend based on "genre" or some feature/tag which might be present already when the item is being introduced. A component of fresh items might be recommended as well to each user (e.g. 10% of his feed).

Better Methods

Graph Machine Learning

- The main point is structuring the dataset in terms of nodes and edges and using proper ML methods on top

- Graph Neural Network, combined with other graph-aware methods, have seen high interest in published papers lately, more expected in the coming years



LightGCN

- ✓ light in terms of weights to store
- ✓ it better exploits the bipartite graph structure by "aggregating" and "weighting" the information coming from broader node neighbourhoods
- ✓ the model performs better than standard WMF, which can be considered as a first order approximation of the original dataset

Thank you !

Davide Roznowicz

droznowi@sissa.it