



SISTEMAS DE RECOMENDACIÓN

Assignment

February 2023

Javier Parapar & Alfonso Landín

Information Retrieval Lab
Computer Science Department
University of A Coruña



Assignment

- **The Spotify Million Playlist Dataset Challenge** consists of a dataset and evaluation to enable research in music recommendations.
- It follows the **ACM RecSys Challenge** 2018 edition.
- This is a dataset of 1 million playlists consist of over 2 million unique tracks by nearly 300,000 artists, and represents the largest public dataset of music playlists in the world.
- Each playlist in the MPD contains a playlist title, the track list (including track IDs and metadata), and other metadata fields (last edit time, number of playlist edits, and more).
- The goal of the challenge is to develop a system for the task of **automatic playlist continuation**.
- Participants have to create **a ranking of 500 recommended candidate tracks** for each target playlist.

Objective

- The objective of the assignment is to **read** and **process** the training playlist, and then **recommend** tracks from the dataset for the test playlist, and finally **evaluate** the results based on a provided set of gold data.
- As the golden truth for the challenge playlists is not provided we will use a modified dataset for training and evaluation:
 - <https://irlab.org/~sr-gced/> (Usuario: user – Password: sr-gced-2023)
 - Training set is the challenge training dataset with 10000 playlists removed
 - Test set is the remaining 10000 playlists, split the same way as the challenge set.
- There are no mandatory technologies, but a proper implementation of the solution is expected.

- (03/03/2023) We need a baseline to compare our results against. Recommending popular items is a simple, non personalized approach that usually gives decent results.
 - You must process the training dataset and determine the most popular tracks.
 - For each playlist in the test dataset you must generate a ranked list of 500 tracks.
 - Tracks already present in the playlist seed must not be presents in the list of recommended tracks.
 - Results must be produced using the format specified in the challenge.
- We need to evaluate our results. We will use the metrics defined in the challenge: R-Precision, NDCG and Recommended Songs clicks.

- (24/03/2023) Neighbourhood-based recommendation. We will try a simple formulation:

$$\hat{r}_{u,i} = \sum_{v \in V_u} s_{u,v} r_{v,i} \quad (1)$$

$$\hat{r}_{u,i} = \sum_{j \in \mathcal{J}_i} s_{i,j} r_{u,j} \quad (2)$$

- Both user-based (Eq. 1) and item-based (Eq. 2)
- V_u represents the neighbourhood of user u . \mathcal{J}_i are the neighbouring items of item i . The size of the neighbourhood is a hyper-parameter k .
- $s_{.,.}$ represents the similarity metric between users or items.
- We will use the cosine similarity both for neighbourhood calculations and for weighting score contributions when obtaining the estimated score.
- How to recommend? For each item not *rated* estimate a score and recommend the top ones.

- (18/04/2023) Matrix Factorization: PureSVD.

$$R = U \times \Sigma \times V^T \simeq \tilde{U} \times \tilde{\Sigma} \times \tilde{V}^T \quad (3)$$

- We will test two variations:


- Compute the decomposition using both training and test input data (all playlists are known when training the model).
- Compute the decomposition using only the training data. For the new playlists we will project the new row into the latent space using track features matrix (V)

$$\vec{u}_{m+1} = \vec{r}_{m+1} \times \tilde{V} \quad (4)$$

- $\vec{r}_{m+1} \equiv$ new user preferences
 - $\vec{u}_{m+1} \equiv$ new user features vector
- Scores for each pair playlist/tracks are computed by the dot product of their respective latent vectors:

$$\hat{r}_{u,i} = \vec{u}_u \times \Sigma \times \vec{v}_i^T \quad (5)$$

- `sparsesvd` 

- (09/05/2023) Other models: `word2vec` [1, 2]
 - Word2vec trains a neural model to predict words in text given their context (CBOW model) or to predict the context of the input word in a sentence (Skip-gram model). These models are able to compute word embeddings that capture both syntactic and semantic information in a lower dimensional space.
- Parapar et al. [3] showed that the recommendation task can be modeled as a query expansion task, where user profiles are analogous to documents and queries, and items are analogous to words.
- We will use this modeling of the problem to compute item embeddings. We will train a `word2vec` model using the playlists as input. Each playlist will be an input document.
- To recommend we will use the same item-based formulation as in iteration 1 (Eq. 2), using the cosine between item embeddings as similarity measure.
- Gensim `word2vec` implementation 

- You are expected to work in **teams of 2** persons
- At the end of the course, every student should be able to explain and modify the work done. Do not over-specialise in a particular part of the task
- Evaluation will be based on both the accomplishment of the requirements and the quality of the project, including its efficiency and effectiveness.



Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean.
Efficient Estimation of Word Representations in Vector Space.
CoRR, abs/1301.3:1–12, 2013.



Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and
Jeffrey Dean.
Distributed Representations of Words and Phrases and Their
Compositionality.
In Proceedings of the 26th International Conference on Neural
Information Processing Systems, NIPS'13, pages 3111–3119,
USA, 2013. Curran Associates Inc.



Javier Parapar, Alejandro Bellogín, Pablo Castells, and Álvaro
Barreiro.
Relevance-based language modelling for recommender systems.
Inf. Process. Manage., 49(4):966–980, July 2013.

 @jparapar  @Dosvelasnegras

<http://www.dc.fi.udc.es/~parapar>