



# Music Recommendation by Unified Hypergraph: Combining Social Media Information and Music Content

Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, Xiaofei He

Presented by Antuan Byalik

# Goal

- Create a music recommendation system that is better than pure user rating
- Incorporate social media information to get better recommendations

# Why Hypergraphs?

- The paper lists two main problems when using social media in this context:
  - “There are many different types of objects and relations in music social communities, which makes it difficult to develop a unified framework taking into account all objects and relations.”
  - “In these communities, some relations are much more sophisticated than pairwise relation, and thus cannot be simply modeled by a graph.”

# Notation and Formal Definition

- $G(V, E, w)$  – denotes a Hypergraph where
  - $V$  – set of vertices,  $E$  – set of hyperedges,  $w$  – weight function
- $D(e)$  – degree of a hyperedge = cardinality of that edge's set
- Unified Hypergraph – Hypergraph with multi-type vertices and hyperedges
  - Vertices/edges represents users/groups/songs

# The Approach

- Focus on six objects and nine relations
- Construct a unified Hypergraph

**Table 1: Objects in our data set.**

Objects	Notations	Count
Users	$U$	2596
Groups	$G$	1124
Tags	$Ta$	3255
Tracks	$Tr$	16055
Albums	$Al$	4694
Artists	$Ar$	371

**Table 2: Relations in our data set.**

Relations	Notations	Count
Friendship relations	$R_1$	4503
Membership relations	$R_2$	1124
Listening relations	$R_3$	304860
Tagging relations on tracks	$R_4$	10936
Tagging relations on albums	$R_5$	730
Tagging relations on artists	$R_6$	36812
Track-album inclusion relations	$R_7$	4694
Album-artist inclusion relations	$R_8$	371
Similarities between tracks	$R_9$	-

# The Approach

- $E_1$  for each pairwise friendship with weight = 1
- $E_2$  with vertices corresponding to all users in a group including the group itself for every group with weight = 1
- $E_3$  for each user-track combination with weight = frequency
- $E_4/E_5/E_6$  for tracks/albums/artists with three vertices in each for user, tag and resource - weight = 1

# The Approach

- $E_7/E_8$  represents album with all its tracks and similarly artist contains all albums by that artist with all weights = 1
- $E_9$  is the hyperedge set for the knn graph on acoustic-based music similarities with the weight = similarity of the two tracks (internally defined formula for this)
- Lastly, construct the vertex-hyperedge incidence matrix

# Methodology

- Offline training – construct the Hypergraph and compute incidence matrix
- Online recommendation – query based on a user and compute rankings
  - Use vertex for user in combination with a defined cost function
  - This gives sorted list of 'closest' results to a particular query



# Conclusions

- Found to work very well on Last.fm and Pandora where data sets were taken and tested
- Similar process could be used for movies and pictures
  - Mentions that in something like facebook the weights might have to be adjusted based on how important social connection is to preference