Universitat de Barcelona

DAS
DATA SCIENCE @ UNIVERSITAT DE BARCELONA

Master on Foundations of Data Science

Recommender Systems

Music Recommnedations

Santi Seguí | 2017-2018

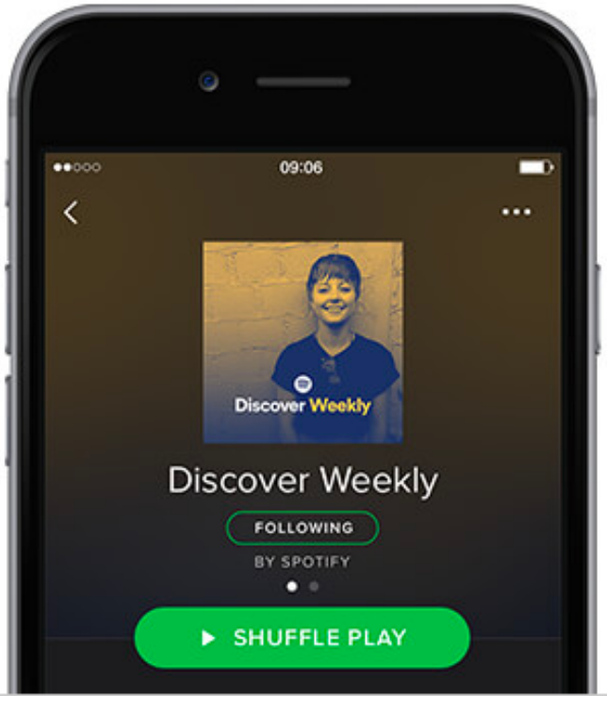
Music Industry

Growing industry

Accelerating transition:

Physical -> digital

Recommendations are now needed!



Not just a format transition, but fundamental revolution. **From ownership towards to access.**

How many songs?

Search/Discovery/Recommendations

- **Query by search**: title, audio fragment, one or more songs
- **Semantic retrieval**: natural language query
- **Based on user profile**: mixed queries
  - Does **context** matter?
    - What are you doing/how are you feeling

“Change of **paradigm** for RecSys: Recommending an **experience**, not just a product/item”

Music Recomendations

Several applications but **still far to be perfect**. Complex problem since users' tastes and musical needs are highly dependent on a **multiple factors**, which are not considered in depth in most of the cases

Online Music Curators

Songza

PANDORA

the echonest

last.fm

Google Play Music

MUSIC

Why recommending music is so difficult?

NETFLIX

Spotify™

NETFLIX

Spotify™

Items

Users

User feedback

User consumption

NETFLIX

Few number of items

Users explicitly rate movies

No Repeated Consumption

Spotify™

More number of items

Music is more niche

Feedback is implicit through streaming behavior

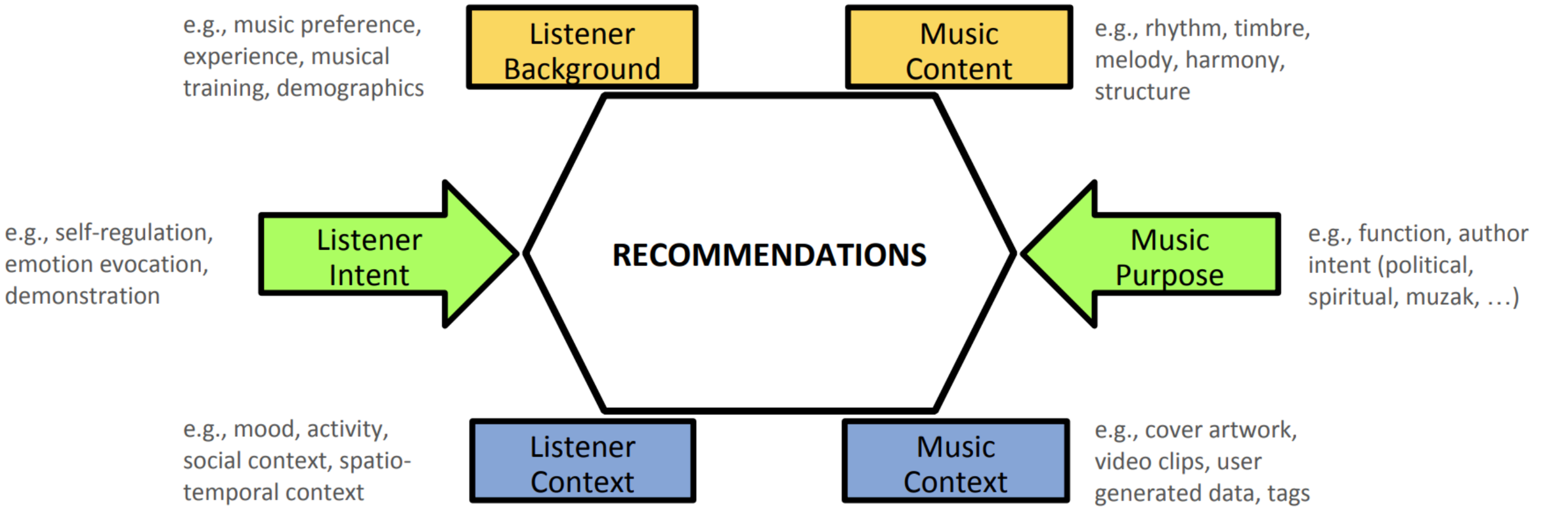
Repeated Consumption

What makes music Recommendation special?

1. Duration of the items (3+ vs. 60+ minutes)
2. Magnitude of the items
3. Sequential consumption
4. Repeated recommendations
5. Consumption behavior often consumed passively (while working, background music, …)
6. Listening Intent and context. Different consumption location/settings: static (e.g., via stereo at home) vs. variable (e.g., via headphones during exercises)
7. Importance of social importance
    1. nitche
8. Highly emotional connoted (in contrast to products.)
9. Music often used for self-expression
10. Various actors for recommendations (listeners, producers, performance, etc,..)
11. Various types of items (songs, albums, artists, audio samples, concerts, venues, fans, etc.)

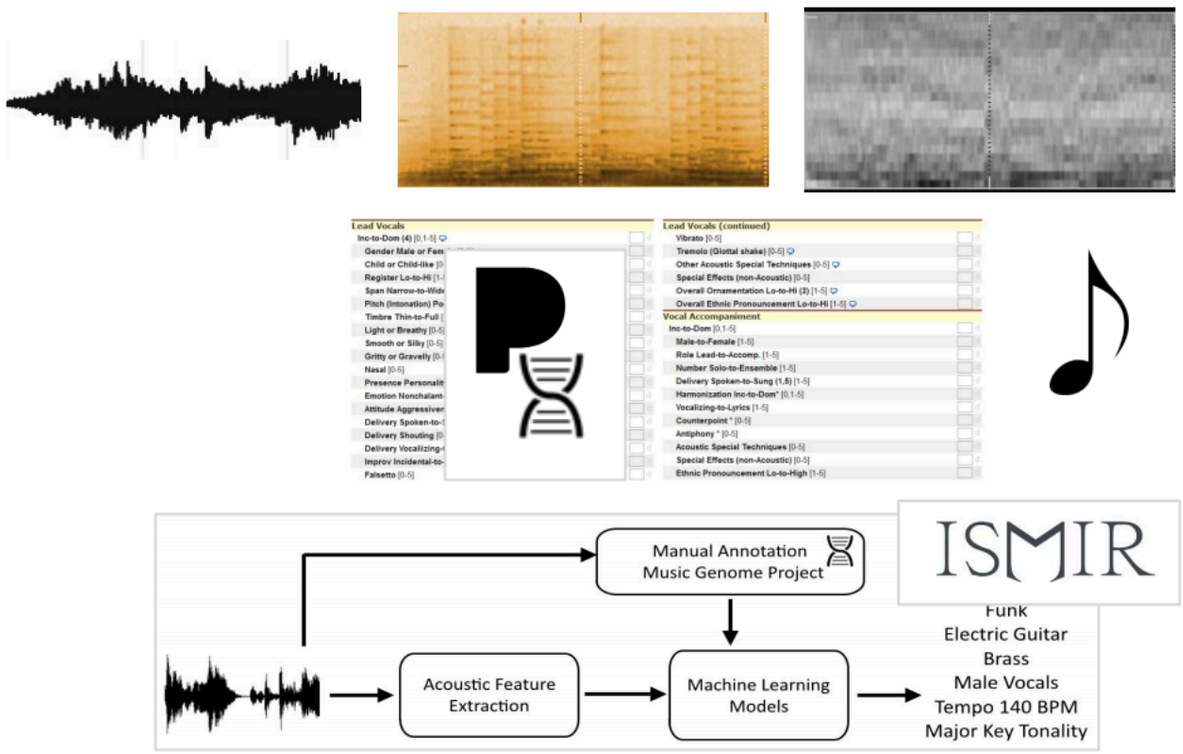
Music Recomendations

multiple factors: intrinsic (**personality** and **emotional state**) ,extrinsic (**activity**) and **contextual information** (weather, social surrounding, location,..) are needed to perform good recommendations


e.g., music preference, experience, musical training, demographics
Listener Background
Music Content
e.g., rhythm, timbre, melody, harmony, structure
e.g., self-regulation, emotion evocation, demonstration
Listener Intent
RECOMMENDATIONS
Music Purpose
e.g., function, author intent (political, spiritual, muzak, …)
e.g., mood, activity, social context, spatio-temporal context
Listener Context
Music Context
e.g., cover artwork, video clips, user generated data, tags

Data?

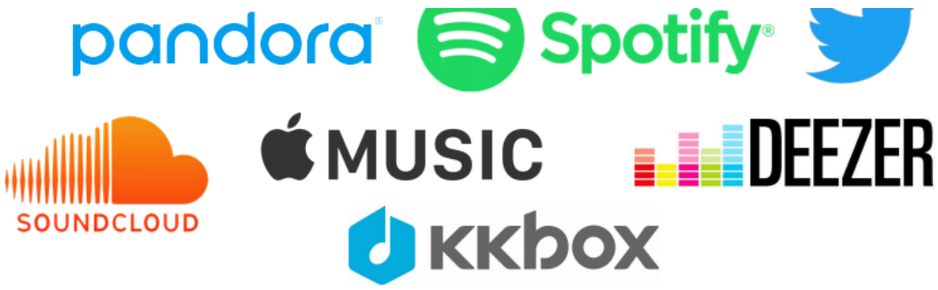
- Content (audio, symbolics, lyrics)
  - Machine content analysis
  - Human labelling
- Meta-data
  - Editorial
  - Curators
  - Multi-modal



Data?

- User-generated
  - “Community meta-data”
  - e.g. reviews, tags

- Interaction data
  - Listening logs
  - Feedback (likes)
  - Purchases



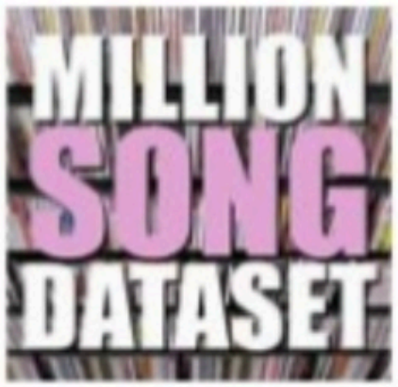
- Curated collections
  - Playlists, radio channels
  - CD album compilations

DataSets

- Million Song Dataset: Meta data for 1.000.000 songs

+ Echo Nest Taste profile subset
Listening data from **1.1m users** for **380k songs**

+ 7digital
Raw audio clips (over 99% of dataset)



https://labrosa.ee.columbia.edu/millionsong/

Methods for recommending Music

Perhaps the best **content based** recommender music systems

Problems?
Features are manually fixed
Small number of item in comparision with other systems

Audio Content Analysis

- **True Content Based Recommendations** (e.g. Pandora)
- Features can be extracted from the audio file
  - no other data or community is needed
  - no cultura biases (no popularity bias, no subjectivity ratings,..)
- Learning of high level descriptors via machine learning
- Deep-Learning is becoming really popular (Temporal representation of sound vida CNN and RNN)

Audio **Content** Analysis

- Beat/downbeat -> tempo 5bpm
- Timbre
- Tonal Features
- **Semantic features** via machine learning (not_danceable, mood_not_happy, mod_focus)

Audio Content Analysis

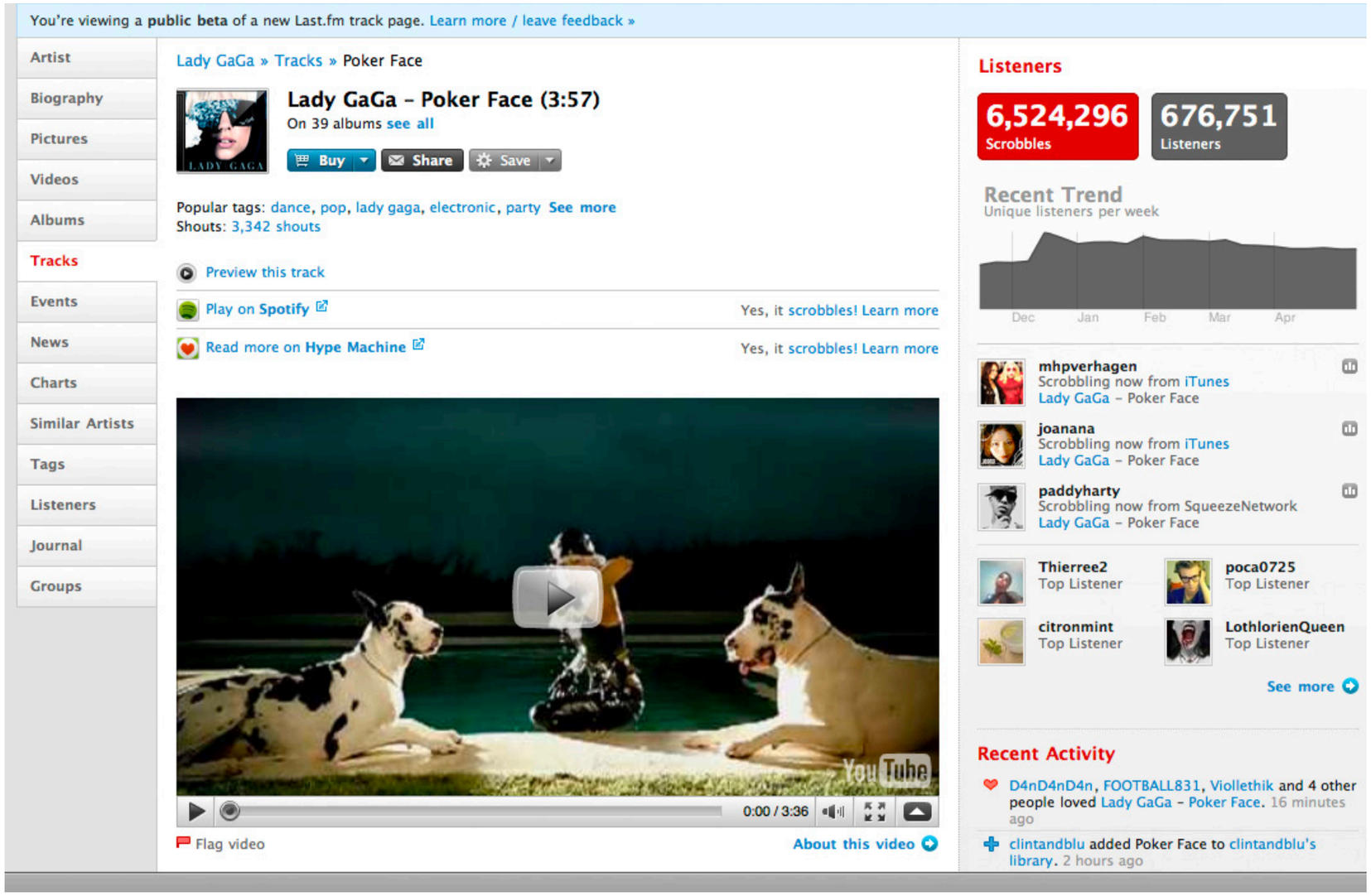
- Essentia (C++, Python): http://essentia.upf.edu/documentation/
- Librosa (Python): https://github.com/librosa
- Madmon (Python): https://github.com/CPJKU/madmom
- Marsyas (C++): http://marsyas.info/
- MIRToolbox (Matlab): https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox
- jMIR (Java): http://jmir.sourceforge.net/

Text Content Analysis

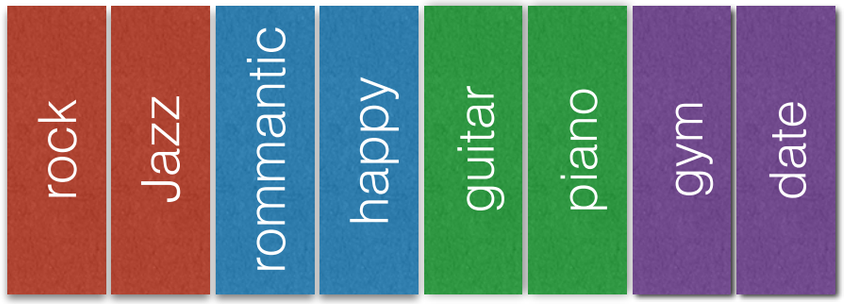
- Text processing of **user-generated** content and **lyrics**
  - Captures aspects beyond pure audio signals
  - no audio file is necessary
  - Generate content features similarly as it is done in text files
    - Bag-of-Words, Vector Space, Tf-Idf
    - Topic Models, word2vect
  - **Sentiment** analysis from reviews, Tag-based similarity, mood detection on lyrics

Text Content Analysis

Yesterday all my troubles seemed so far away.
Now it looks as though they're here to stay.
Oh, I believe in yesterday.
Suddenly, I'm not half the man I used to be.
There's a shadow hanging over me.
Oh, yesterday came suddenly.
Why she had to go, I don't know,
She wouldn't say.
I said something wrong,
Now I long for yesterday.
Yesterday love was such an easy game to play.
Now I need a place to hide away.
Oh, I believe in yesterday.
Why she had to go, I don't know,
She wouldn't say.
I said something wrong,
Now I long for yesterday.
Yesterday love was such an easy game to play.
Now I need a place to hide away.
Oh, I believe in yesterday.



Content Based-Approach



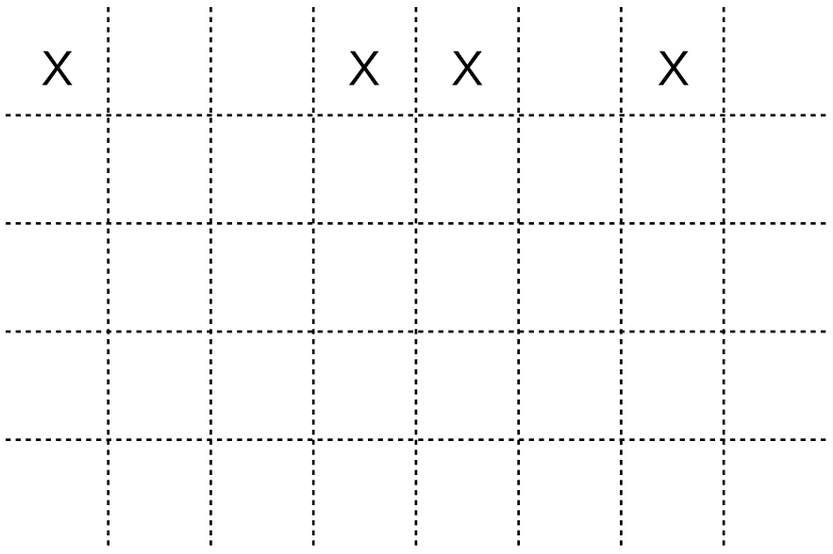
Automatic Tagging

Genre

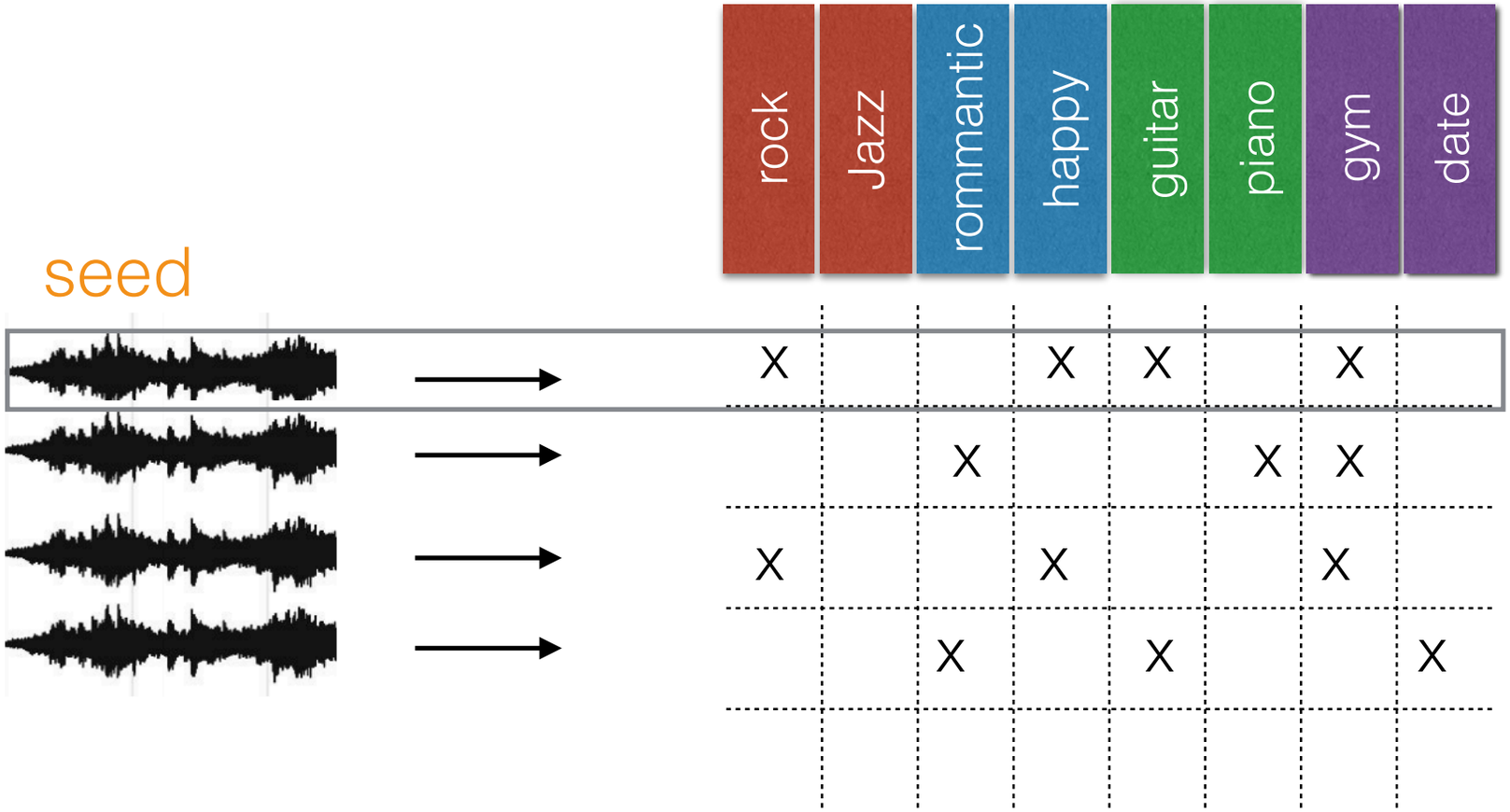
mood

intrument

usage



Content Based-Approach



Collaborative Filtering

- Exploiting interaction data
  - Usually closer of what the user wants
  - Implicit (plays) and explicit (thumbs) data is used
  - Task: complete the rating matrix using SVD or other matrix factorization tecniques

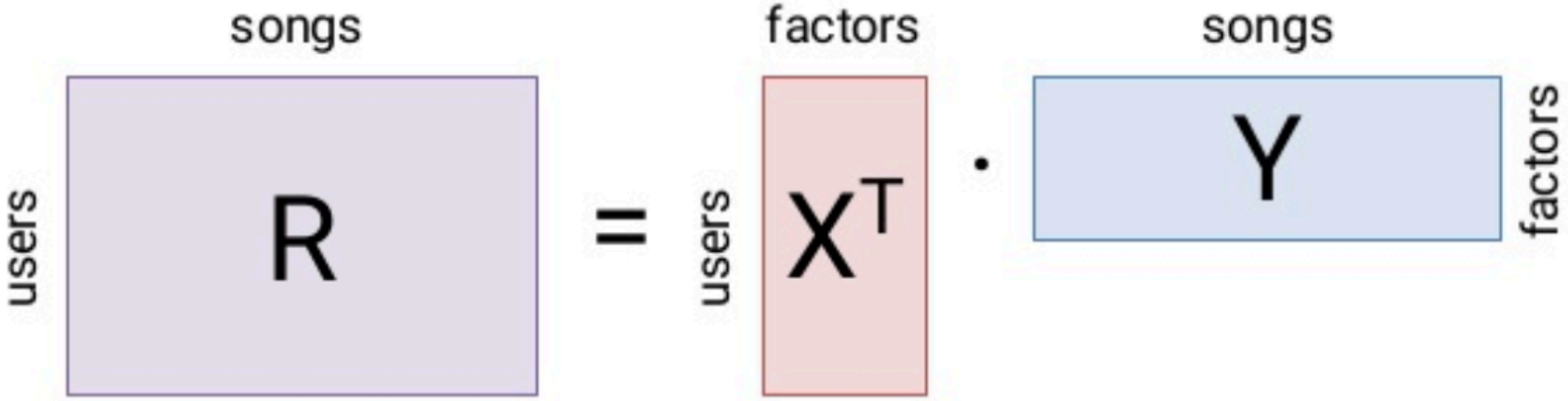
$$b_{ui} = \mu + b_{u,type(i)} + b_{u,session(i,u)} + b_i + b_{album(i)} + b_{artist(i)} + \frac{1}{|genres(i)|} \sum_{g \in genres(i)} b_g + c_i^T f(t_{ui})$$

Cold Start

- Collaborative Filtering methods fails on new item
- Solution:

Deep content-based music recommendation A Van den Oord, S Dieleman, B Schrauwen Advances in neural information processing systems, 2643-2651	356	2013

  - Predict the latent factors from music data when they cannot be obtained from usage data



listening data
play counts

user profiles
latent factors

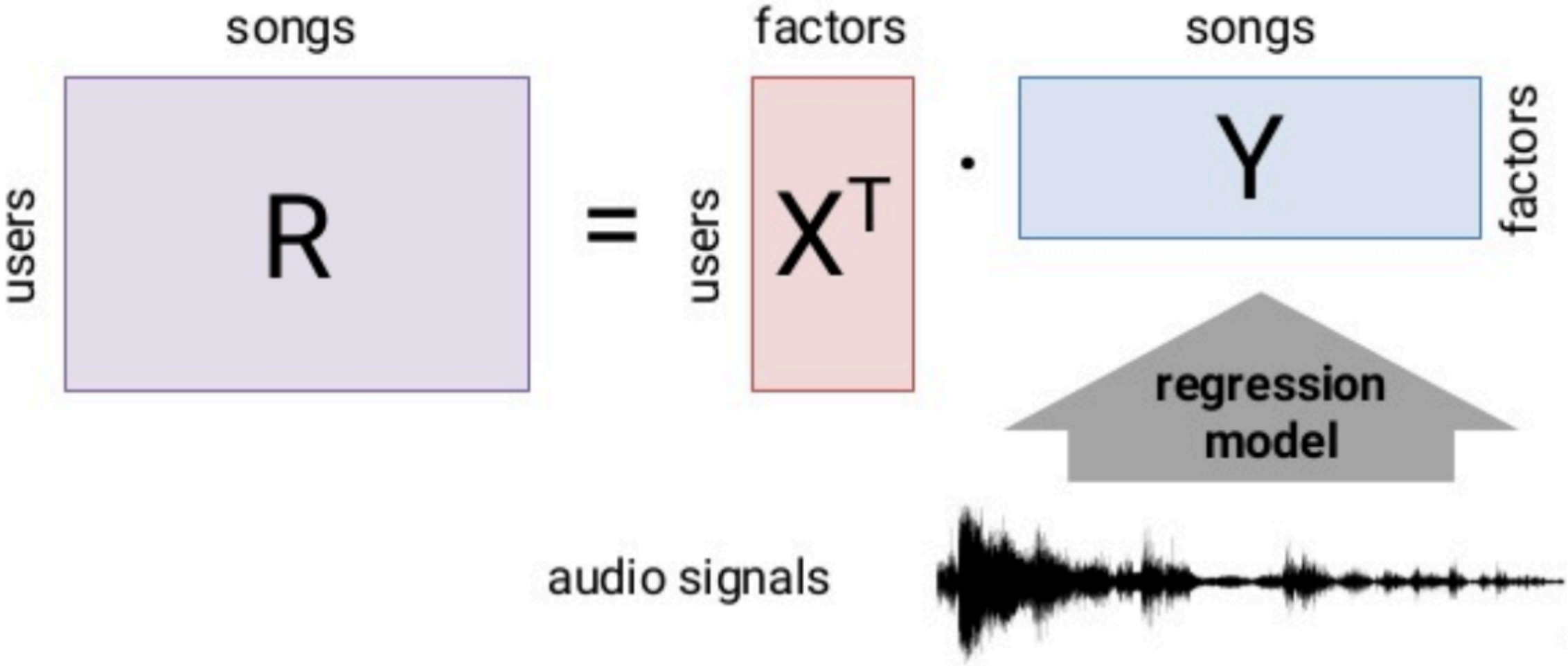
song profiles
latent factors

$$
\begin{aligned}
p_{ui} &= I(r_{ui} > 0), \\
c_{ui} &= 1 + \alpha \log(1 + \epsilon^{-1} r_{ui}).
\end{aligned}
$$

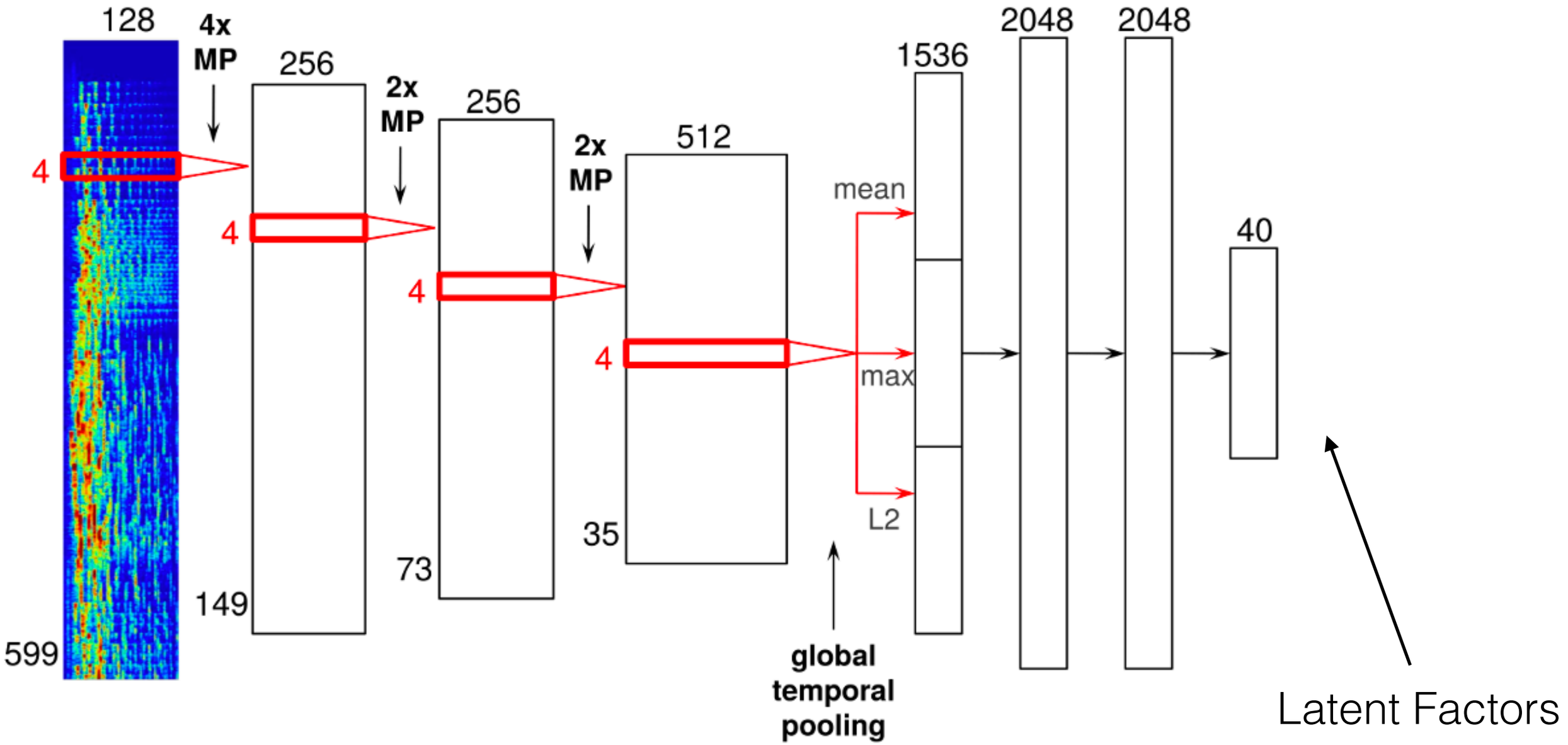
$$
\min_{x_\star, y_\star} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u ||x_u||^2 + \sum_i ||y_i||^2 \right)
$$

Deep content-based music recommendation 356 2013
A Van den Oord, S Dieleman, B Schrauwen
Advances in neural information processing systems, 2643-2651

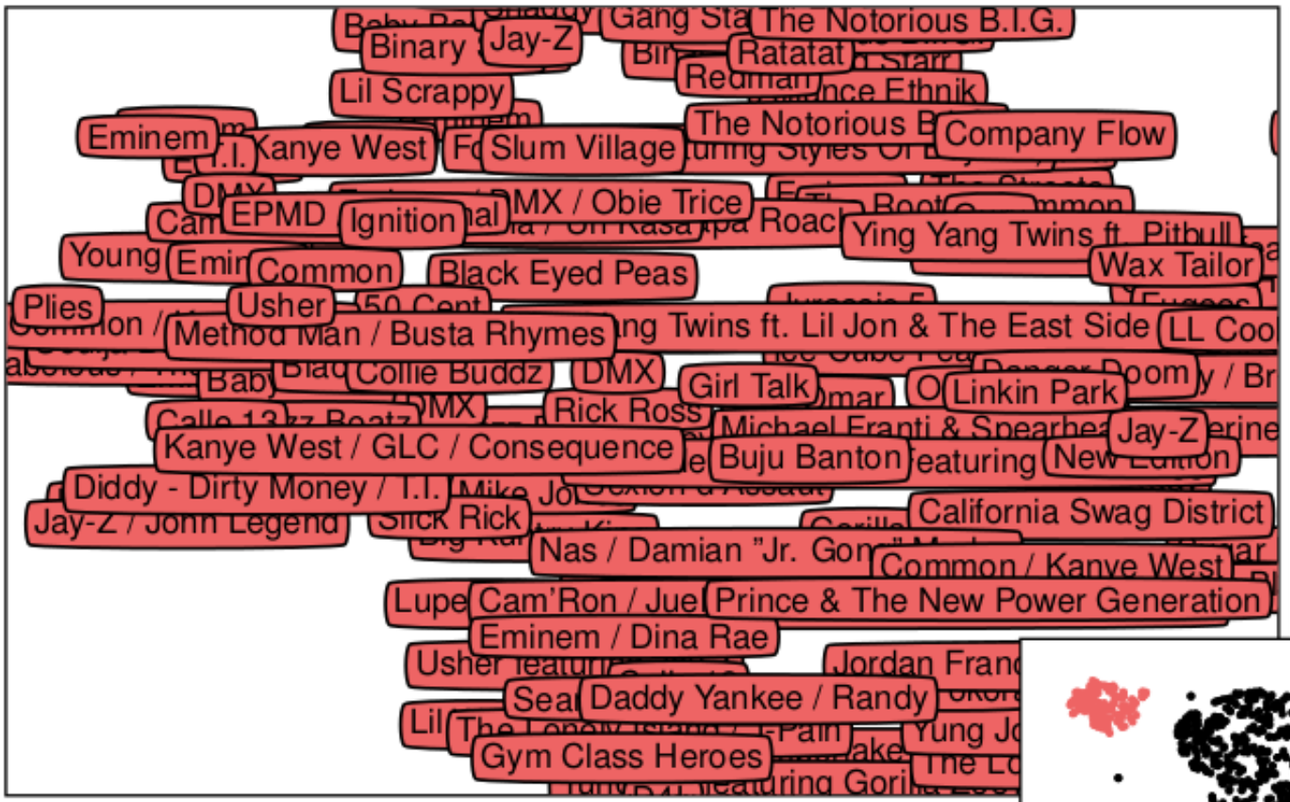
Learn from audio the latent topics

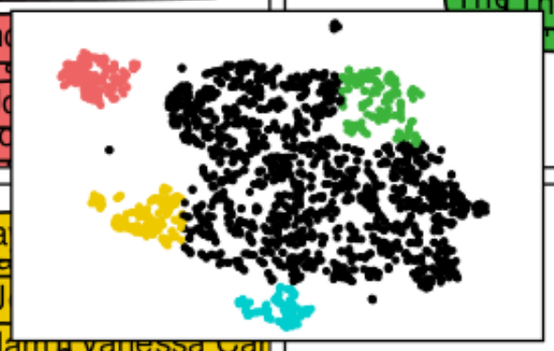


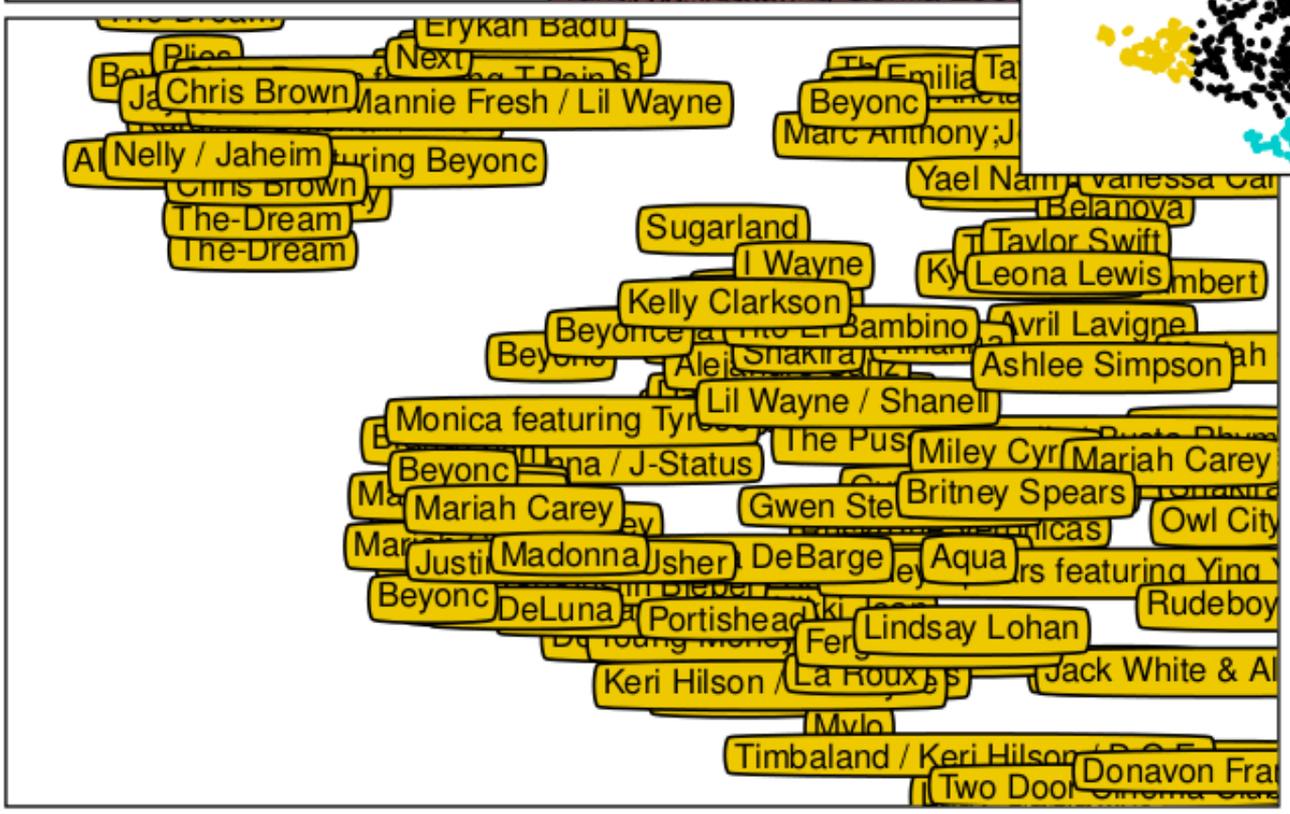
Raw audio models

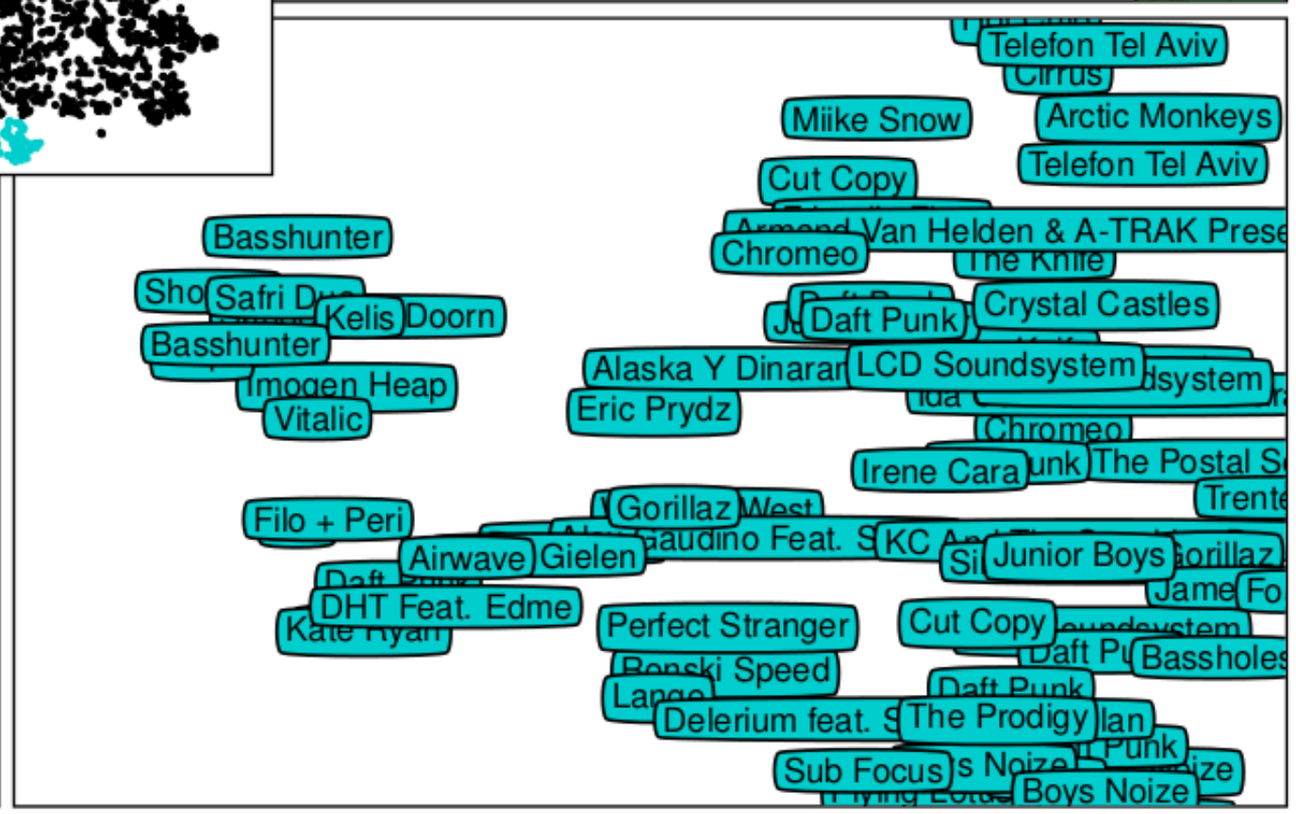


Deep content-based music recommendation 356 2013
A Van den Oord, S Dieleman, B Schrauwen
Advances in neural information processing systems, 2643-2651









Deep content-based music recommendation 356 2013
A Van den Oord, S Dieleman, B Schrauwen
Advances in neural information processing systems, 2643-2651

Other important issues

Temporal Aspect

- Recommending next tracks: **Temporal ordering matters!!**
- Notion of music rotation
    - Popularity categories
    - Music attributes
    - Sound attributes
    - Artist separation
- Predict the **best time** for the next user interaction for the item
- Modelling transitions in listeining habits

A good recommendation?

What makes a good recommendation?

- Accuracy
- A good balance of:
  - Novelty vs. familiarity/popularity
  - Diversity vs. similarity
- Transparency / Interpretability
- Listener context

← Influential Factors?

- Listener
- Musical anchor
- Mood

Remember: It's about **recommending an experience**!!

[Celma, 2010] Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space, Springer
[Amatriain, Basilico, 2016] Past, Present, and Future of Recommender Systems: An Industry Perspective, RecSys

Accuracy (is not enough)

- Too much focus on **accuracy** —> biases (i.e. popularity and similarity based)
- Tradeoff popularity vs. personalization
- Particular risk of selection bias when RecSys is the oracle
- **RMSE**?? (as in Netflix) just one side of the recommendations

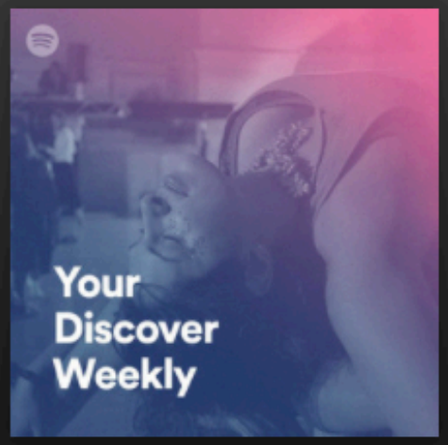
Novelty

- Introducing novelty to balance against popularity (or familiarity) bias.
- Listeners whats to hear what is hype, but they also need their dose of novelty… Once a while.
    - How far novel?
    - How often?
    - When?

	“Yep, novelty is fine”	**“No novelty, please!”**
Listener	Jazz musician	My mother
Musical anchor	Exploring a new friend’s music library	Self-made playlist
Focus	Discovery	Craving for my hyper-personalized stuff

Search

Sophia Ciocca



MADE FOR SOPHIA

Discover Weekly

Your weekly mixtape of fresh music. Enjoy new discoveries and deep cuts chosen just for you. Updated every Monday, so save your favourites!

Made for **Sophia Ciocca** by **Spotify** • 30 songs, 2 hr 3 min

PLAY FOLLOWING ···

FOLLOWER
1

Filter

Download

	TITLE	ARTIST	ALBUM	
+	To Hugo	Clogs	The Creatures In Th...	2 days ago
+	Little Worlds	Mandolin Orange	Such Jubilee	2 days ago
+	Quiet Voices	Mike Vass	In the Wake of Neil ...	2 days ago
+	Sometimes	Goldmund	Sometimes	2 days ago
+	Sileo	Rhian Sheehan	Stories From Elsewh..	2 days ago
+	Hollow Home Rd	Brolly	Hollow Home Rd	2 days ago
+	Marigold	Mother Falcon	You Knew	2 days ago
+	Things Happen	Dawes	All Your Favorite Ba...	2 days ago
+	Sliding Down	Edgar Meyer, Mike ...	The Best of Edgar M..	2 days ago
+	Celeste	Pete Kuzma	Equilibrium	2 days ago

Fee +

3:43 5:49

Spotify Architecture

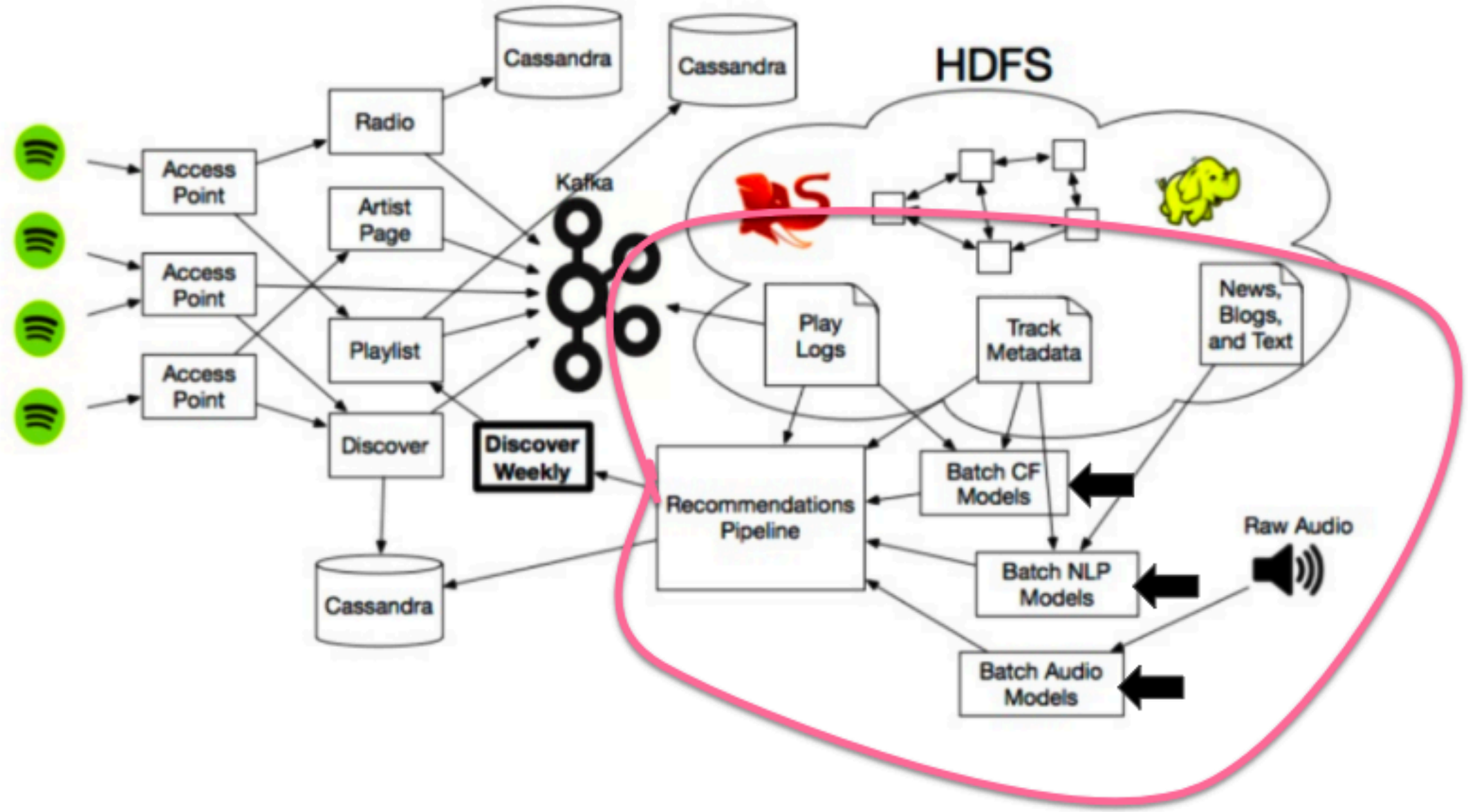


Image credit: Chris Johnson, Spotify

How does it work?

- Spotify, to Discover Weekly, employs three main types of recommendation models:
  - **Collaborative Filtering** models (i.e. the ones that Last.fm originally used), which work by analyzing *your* behavior and *others'* behavior.
  - **Natural Language Processing (NLP)** models, which work by analyzing *text*.
  - **Audio** models, which work by analyzing the *raw audio tracks themselves*.

CF in Spotify

- Unlike Netflix, though, Spotify doesn't have those stars with which users rate their music. Instead, Spotify's data is **implicit feedback**—specifically, the **stream counts** of the tracks we listen to, as well as additional streaming data, including whether a user saved the track to his/her own playlist, or visited the Artist page after listening.

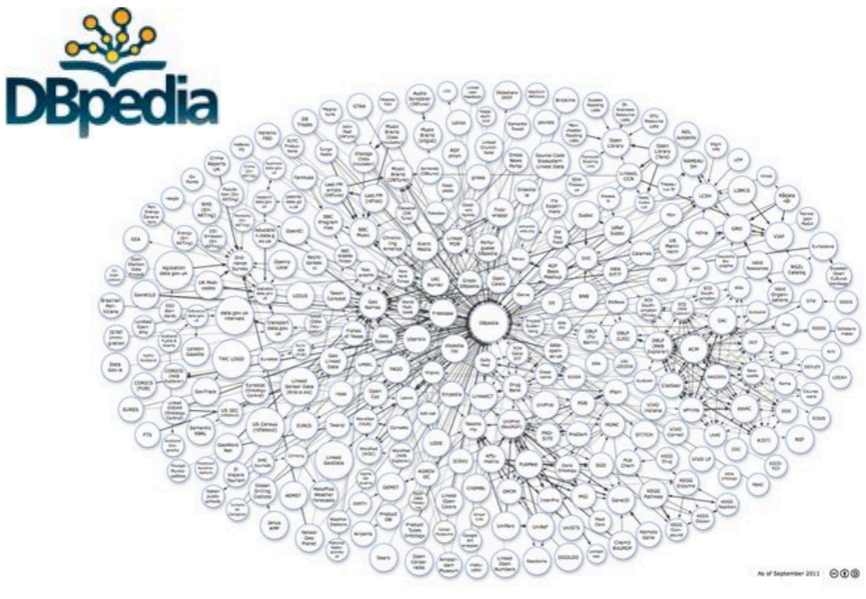
- **Each row represents one of Spotify's 140 million users** and **each column represents one of the 30 million songs** in Spotify's database.

CF in Spotify

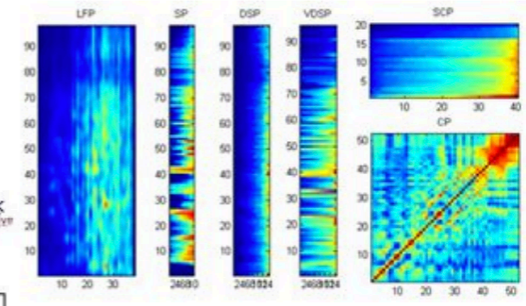
- Matrix factorization using an optimization strategy

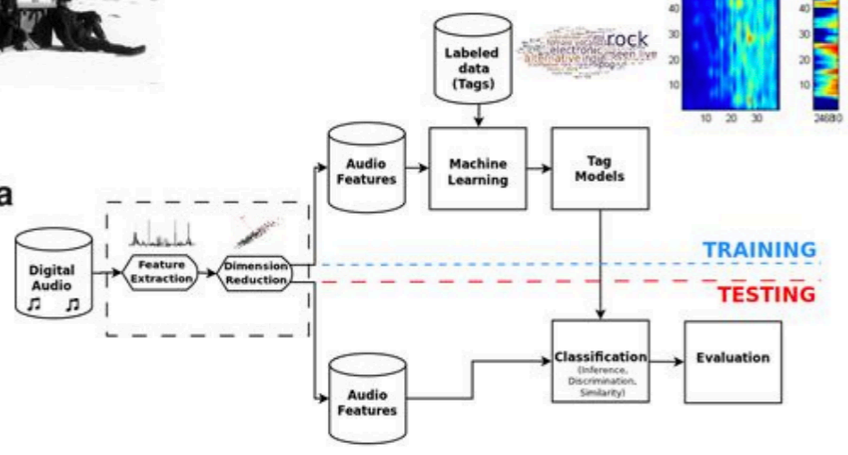
- When it finishes, we end up with two types of vectors, represented here by X and Y. **X is a *user* vector**, representing one single user's taste, and **Y is a *song* vector**, representing one single song's profile.

Hybrid MRS fusing knowledge-based recommendations and audio content-based recommendations obtained via auto-tagging (rank fusion)

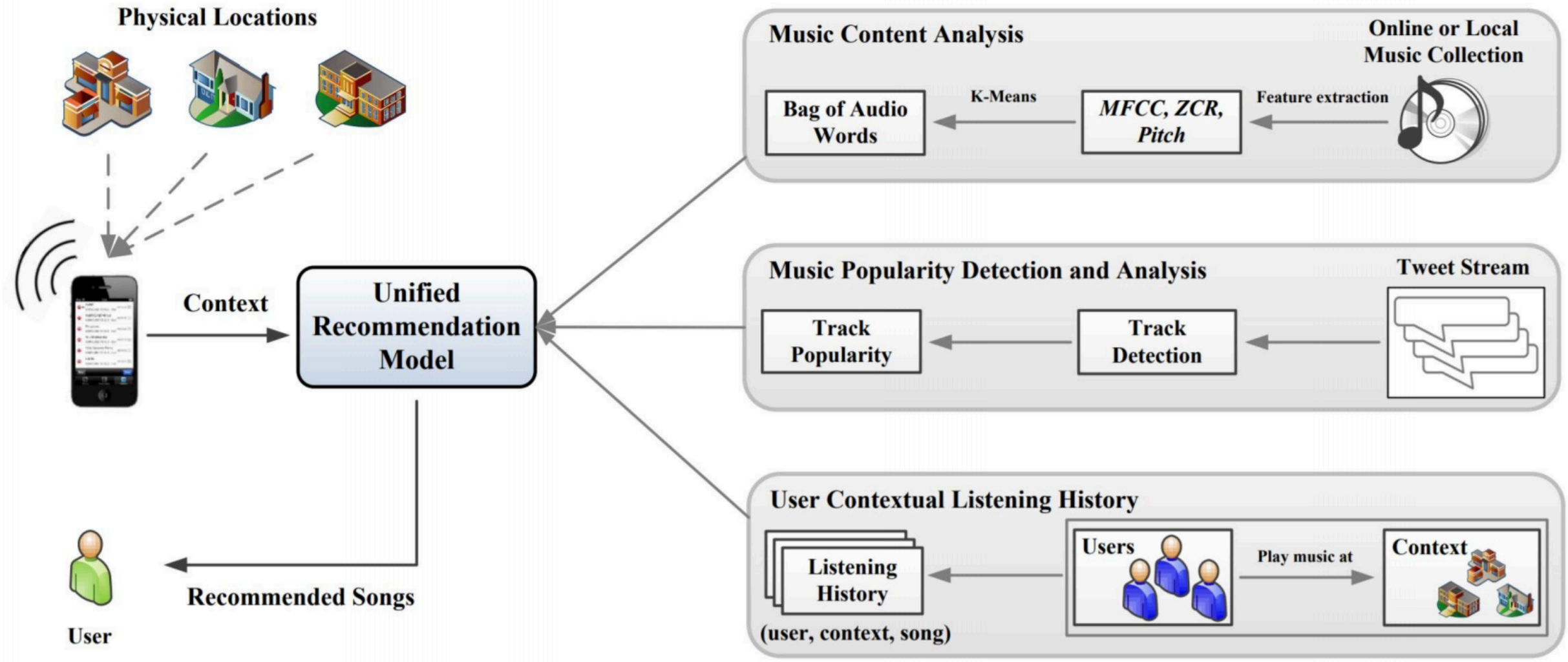


+





Location-aware music recommendation using auto-tagging and hybrid matching 59 2013
M Kaminskas, F Ricci, M Schedl
Proceedings of the 7th ACM conference on Recommender systems, 17-24



Just-for-me: An adaptive personalization system for location-aware social music recommendation — 52 — 2014
Z Cheng, J Shen
Proceedings of international conference on multimedia retrieval, 185

RecSys Challenge 2018

Welcome ACM RecSys Community! For this year's challenge, use the Spotify Million Playlist Dataset to help users create and extend their own playlists.

Read on for all the details. Good luck!

The RecSys Challenge 2018 is organized by Spotify, The University of Massachusetts, Amherst, and Johannes Kepler University, Linz.

Have a question, suggestion or concern? Let us know by emailing us at recsyschallenge@spotify.com

International Journal of Multimedia Information Retrieval (2018) 7:95–116
https://doi.org/10.1007/s13735-018-0154-2

TRENDS AND SURVEYS

Current challenges and visions in music recommender systems research

Markus Schedl[1] · Hamed Zamani[2] · Ching-Wei Chen[3] · Yashar Deldjoo[4] · Mehdi Elahi[5]

Received: 28 September 2017 / Revised: 17 March 2018 / Accepted: 27 March 2018 / Published online: 5 April 2018

Abstract
Music recommender systems (MRSs) have experienced a boom in recent years, thanks to the emergence and success of online streaming services, which nowadays make available almost all music in the world at the user's fingertip. While today's MRSs considerably help users to find interesting music in these huge catalogs, MRS research is still facing substantial challenges. In particular when it comes to build, incorporate, and evaluate recommendation strategies that integrate information beyond simple user–item interactions or content-based descriptors, but dig deep into the very essence of listener needs, preferences, and intentions, MRS research becomes a big endeavor and related publications quite sparse. The purpose of this trends and survey article is twofold. We first identify and shed light on what we believe are the most pressing challenges MRS research is facing, from both academic and industry perspectives. We review the state of the art toward solving these challenges and discuss its limitations. Second, we detail possible future directions and visions we contemplate for the further evolution of the field. The article should therefore serve two purposes: giving the interested reader an overview of current challenges in MRS research and providing guidance for young researchers by identifying interesting, yet under-researched, directions in the field.

Keywords Music recommender systems · Challenges · Automatic playlist continuation · User-centric computing

Current Research Challenges

1) **Cold Start problem**

2) **Automatic playlist continuation**

3) **Evaluation of Music Recommender Systems**

1) Cold Start Problem

- High sparsity: >99% in most of the cases.
- Cold-Start problem on items and users
- Solutions:
  - items: could be solve using a CB (using acoustic features) or Hybrid methods (CB + CF)
  - Cross domain recommendations
  - Active learning methods. Identify and elicit data that can represent the preferences of users better than what that they provide themselves

2) Automatic playlist continuation

- Automatic playlist generation (APG) refers to the automatic creation of sequences of tracks
- Infer the intended purpose of a given playlist
- Different approaches:
    - target chracteristics of the playlist are specified as multiple contraints, as for instance musical attributes or metadata such as artist, tempo and styel
    - single seed track
    - start and an end track
- build the background knowledge of the music catalog for playlist generation from manually-curated playslists

3) Evaluation of Music Continuation

- How to asses the quality of a playlist?
- Accuracy, Precision, Recall and other error mesures (between predicted and true ratings) are the most common mesures in order to evaluate the quality of the recommender.

The Task

The goal of the challenge is to develop a system for the task of automatic playlist continuation. Given a set of playlist features, participants' systems shall generate a list of recommended tracks that can be added to that playlist, thereby 'continuing' the playlist. We define the task formally as follows:

Input

A **user-created** playlist, represented by

- Playlist metadata (see the dataset README),
- *K* seed tracks: a list of the *K* tracks in the playlist, where K can equal 0, 1, 5, 10, 25, or 100.

Output

- A list of 500 recommended candidate tracks, ordered by relevance in decreasing order.
- Note that the system should also be able to cope with playlists for which no initial seed tracks are given. To assess the performance of a submission, the output track predictions are compared to the ground truth tracks ("reference set") from the original playlist.

Metrics

Submissions will be evaluated using the following metrics. All metrics will be evaluated at both the track level (exact track must match) and the artist level (any track by that artist is a match). In the following, we denote the ground truth set of tracks by G, and the ordered list of recommended tracks by R. The size of a set or list is denoted by $|\cdot|$, and we use from:to-subscripts to index a list. In the case of ties on individual metrics, earlier submissions are ranked higher.

R-precision

R-precision is the number of retrieved relevant tracks divided by the number of known relevant tracks (i.e., the number of withheld tracks):

$$\text{R-precision} = \frac{|G \cap R_{1:|G|}|}{|G|}.$$

The metric is averaged across all playlists in the challenge set. This metric rewards total number of retrieved relevant tracks (regardless of order).

Normalized discounted cumulative gain (NDCG)

Discounted cumulative gain (DCG) measures the ranking quality of the recommended tracks, increasing when relevant tracks are placed higher in the list. Normalized DCG (NDCG) is determined by calculating the DCG and dividing it by the ideal DCG in which the recommended tracks are perfectly ranked:

$$DCG = rel_1 + \sum_{i=2}^{|R|} \frac{rel_i}{\log_2 i}.$$

The ideal DCG or IDCG is, on our case, equal to:

$$IDCG = 1 + \sum_{i=2}^{|G \cap R|} \frac{1}{\log_2 i}.$$

If the size of the set intersection of G and R, is empty, then the DCG is equal to 0. The NDCG metric is now calculated as:

$$NDCG = \frac{DCG}{IDCG}.$$

Recommended Songs clicks

Recommended Songs is a Spotify feature that, given a set of tracks in a playlist, recommends 10 tracks to add to the playlist. The list can be refreshed to produce 10 more tracks. Recommended Songs clicks is the number of refreshes needed before a relevant track is encountered. It is calculated as follows:

$$\text{clicks} = \left\lfloor \frac{\arg\min_i\{R_i : R_i \in G|\} - 1}{10} \right\rfloor$$

If the metric does not exist (i.e. if there is no relevant track in R), a value of 51 is picked (which is 1 + the maximum number of clicks possible).

Rank aggregation

Final rankings will be computed by using the Borda Count election strategy. For each of the rankings of *p* participants according to R-precision, NDCG, and Recommended Songs clicks, the top ranked system receives *p* points, the second system receives *p-1* points, and so on. The participant with the most total points wins. In the case of ties, we use top-down comparison: compare the number of 1st place positions between the systems, then 2nd place positions, and so on.

Spotify RecSys Challenge 2018

Final ranking for the main track

This leaderboard was computed on July 18, 2018 and shows the scores of final leaderboard for the **main** track. It takes into account the final submission scores evaluated against 100% of the challenge set, and rule compliance. Congratuations to the winners and to all that participated.

#	TEAM	RPREC	RPREC RANK	NDCG	NDCG RANK	CLICKS	CLICKS RANK	BORDA	CODE
1	vl6	0.22413802335	1	0.394622225334	1	1.7839	2	329	Code
2	hello world!	0.223353092072	2	0.393233606092	2	1.8952	6	323	Code
3	Avito	0.215316047292	6	0.384550795868	4	1.7818	1	322	Code
4	Creamy Fireflies	0.220195599368	3	0.385686877502	3	1.9335	7	320	Code
4	MIPT_MSU	0.216742914896	4	0.382332403131	5	1.8754	4	320	Code
6	HAIR	0.216291058889	5	0.380252849031	6	2.1815	13	309	Code
7	KAENEN	0.209063981338	11	0.374719751318	8	2.054	10	304	Code
7	BachPropagate	0.20897465397	12	0.373969266315	12	1.8834	5	304	Code
9	Definitive Turtles	0.208598676943	13	0.375079439377	7	2.0781	11	302	Code
10	IN3PD	0.207761375412	14	0.371300599524	14	1.9517	8	297	Code
11	cocoplaya	0.201610155616	18	0.364507318644	16	1.8674	3	296	Code

THE WINNER

Two-stage Model for Automatic Playlist Continuation at Scale

Maksims Volkovs
Layer6 AI
maks@layer6.ai

Himanshu Rai
Layer6 AI
himanshu@layer6.ai

Zhaoyue Cheng
Layer6 AI
joey@layer6.ai

Ga Wu
Vector Institute
wuga@mie.utoronto.ca

Yichao Lu
University of Toronto
yichao@cs.toronto.edu

Scott Sanner
Vector Institute
ssanner@mie.utoronto.ca

ABSTRACT

Automatic playlist continuation is a prominent problem in music recommendation. Significant portion of music consumption is now done online through playlists and playlist-like online radio stations. Manually compiling playlists for consumers is a highly time consuming task that is difficult to do at scale given the diversity of tastes and the large amount of musical content available. Consequently, automated playlist continuation has received increasing attention recently [1, 7, 11]. The 2018 ACM RecSys Challenge [14] is dedicated to evaluating and advancing current state-of-the-art in automated playlist continuation using a large scale dataset released by Spotify. In this paper we present our approach to this challenge. We use a two-stage model where the first stage is optimized for fast retrieval, and the second stage re-ranks retrieved candidates maximizing the accuracy at the top of the recommended list. Our team vl6 achieved 1'st place in both main and creative tracks out of over 100 teams.

of playlist continuation models. At the core of this challenge is the Million Playlist Dataset (MPD) released by Spotify [14]. MPD is the largest publicly available dataset of it's kind with 1M playlists and over 2.2M songs. The challenge task is to create a playlist continuation model that is able to accurately recommend next songs for each playlist.

The models are evaluated using a separate set of 10K test playlists for which a subset of songs are withheld. Notably, test playlists vary significantly in length from 0 songs (cold start) to 100 songs. This simulates the production scenario where recommendation model has to perform well during all phases of playlist creation. To avoid over fitting, test held out songs are not released, and teams are required to submit their recommendations to the evaluation server. Over 100 teams participated in the main track of this challenge and our team vl6 achieved 1'st place in both main and creative tracks.

Our approach is based on a two-stage architecture. The first stage focuses on reducing the large 2.2M song search space to a much

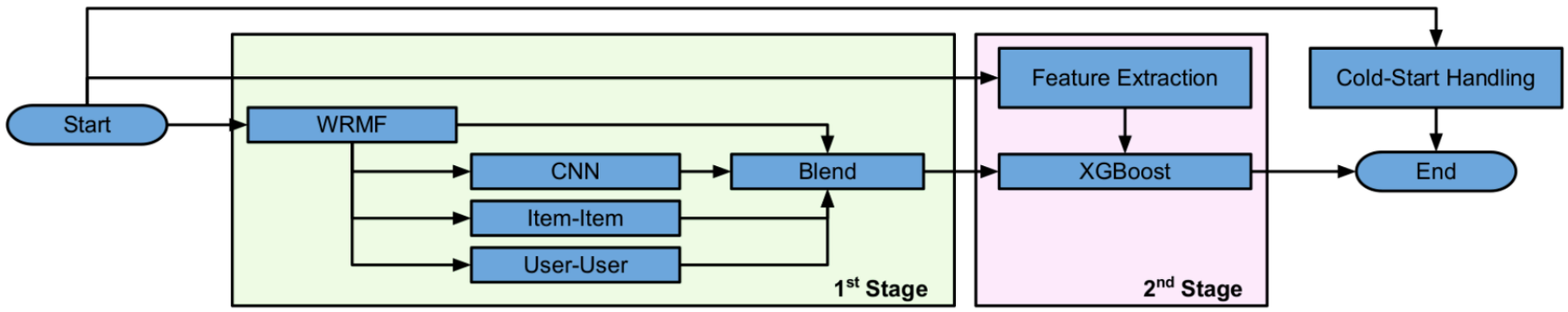


Figure 1: Two-stage model architecture. In the first stage, WRMF is used to retrieve 20K songs for each playlist. CNN, Item-Item and User-User models are then applied to each retrieved song. All model scores together with their linear weighted combination (Blend) are concatenated with extracted playlist-song features and used as input to the second stage. In the second stage, gradient boosting model re-ranks all retrieved songs and outputs the final ranking. Cold start playlists are handled separately (see Section 2.3).

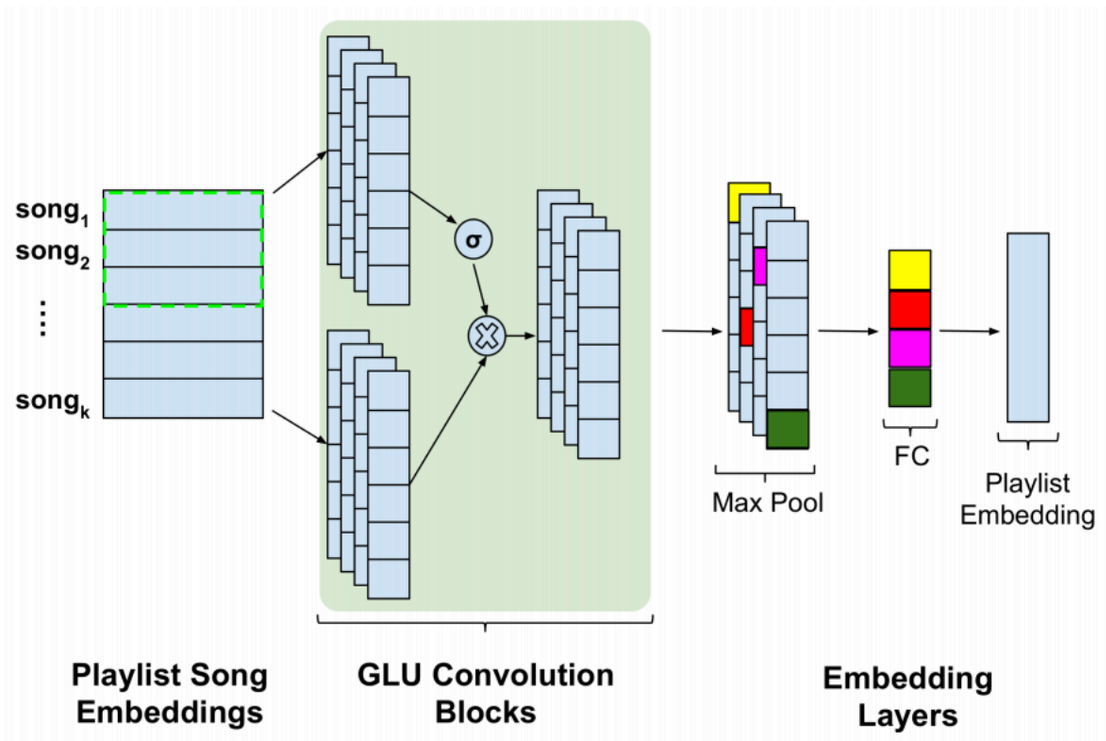


Figure 2: CNN playlist embedding model architecture.

	RPREC	NDCG	Clicks
WRMF	0.1641	0.3350	2.1230
CNN	0.1594	0.3230	2.1157
Item-Item	0.1772	0.3534	2.1649
User-User	0.1768	0.3550	1.6332
Blend	0.1866	0.3728	1.4064
2^{nd} Stage	**0.1985**	**0.3846**	**1.1998**
Cold Start			
Popular	0.0395	0.0815	17.2662
Name SVD	0.1106	0.2083	7.9609

Table 1: Validation set results.

Weighted Regularized Matrix Factorization (WRFM)

Collaborative Filtering for Implicit Feedback Datasets

Yifan Hu
AT&T Labs – Research
Florham Park, NJ 07932

Yehuda Koren*
Yahoo! Research
Haifa 31905, Israel

Chris Volinsky
AT&T Labs – Research
Florham Park, NJ 07932

Abstract

A common task of recommender systems is to improve customer experience through personalized recommendations based on prior implicit feedback. *These systems passively track different sorts of user behavior, such as purchase history, watching habits and browsing activity, in order to model user preferences. Unlike the much more extensively researched* explicit feedback*, we do not have any direct input from the users regarding their preferences. In particular, we lack substantial evidence on which products consumer dislike. In this work we identify unique properties of implicit feedback datasets. We propose treating the data as indication of positive and negative preference associated with vastly varying confidence levels. This leads to a factor model which is especially tailored for implicit feedback recommenders. We also suggest a scalable optimization procedure, which scales linearly with the data size. The algorithm is used successfully within a recommender system for television shows. It compares favorably with well tuned implementations of other known methods. In addition, we offer a novel way to give explanations to recommendations given by this factor model.*

tent based approach creates a profile for each user or product to characterize its nature. As an example, a movie profile could include attributes regarding its genre, the participating actors, its box office popularity, etc. User profiles might include demographic information or answers to a suitable questionnaire. The resulting profiles allow programs to associate users with matching products. However, content based strategies require gathering external information that might not be available or easy to collect.

An alternative strategy, our focus in this work, relies only on past user behavior without requiring the creation of explicit profiles. This approach is known as *Collaborative Filtering* (CF), a term coined by the developers of the first recommender system - Tapestry [8]. CF analyzes relationships between users and interdependencies among products, in order to identify new user-item associations. For example, some CF systems identify pairs of items that tend to be rated similarly or like-minded users with similar history of rating or purchasing to deduce unknown relationships between users and items. The only required information is the past behavior of users, which might be their previous transactions or the way they rate products. A major appeal of CF is that it is domain free, yet it can address aspects of the data

2n Place

TrailMix: An Ensemble Recommender System for Playlist Curation and Continuation

Xing Zhao, Qingquan Song, James Caverlee, and Xia Hu
Texas A&M University

ABSTRACT

This paper describes TrailMix, an ensemble model designed to tackle the RecSys Challenge 2018 for automatic music playlist continuation. TrailMix combines three different models designed to exploit complementary aspects of playlist recommendation: (i) CC-Title, a cluster-based approach for playlist titles; (ii) DNCF, an extension of Neural Collaborative Filtering for taking advantage of the flat interaction among tracks; and (iii) C-Tree, a hierarchical approach akin to Phylogenetic trees for finding relationships between tracks.

KEYWORDS

Recommender System, Playlist Continuation, Constructed Tree Comparison, Neural Network, Collaborative Filtering

ACM Reference Format:
Xing Zhao, Qingquan Song, James Caverlee, and Xia Hu. 2018. TrailMix: An Ensemble Recommender System for Playlist Curation and Continuation. In *Proceedings of the ACM Recommender Systems Challenge 2018 (RecSys Challenge '18), October 2, 2018, Vancouver, BC, Canada.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3267471.3267479

Table 1: Dataset Statistics

Items	Quantity	Proportion
Playlists	1,000,000	
unique tracks	2,262,292	100%
unique tracks (freq ≥ 5)	599,341	96.05%
unique tracks (freq ≥ 100)	70,229	80.67%
unique albums	734,684	
unique artists	295,860	

- Finally, the third model – C-Tree – explores the hierarchical structures of a playlist (e.g., playlist-artist-album-track) akin to Phylogenetic trees for finding relationships between tracks.

Together, Trailmix ensembles these three models toward tackling the RecSys Challenge. In the following, we briefly describe the challenge setting and then dive into the details of each approach[1].