Master on Foundations of Data Science

Recommender Systems

Course Presentation

Santi Seguí | 2017-2018

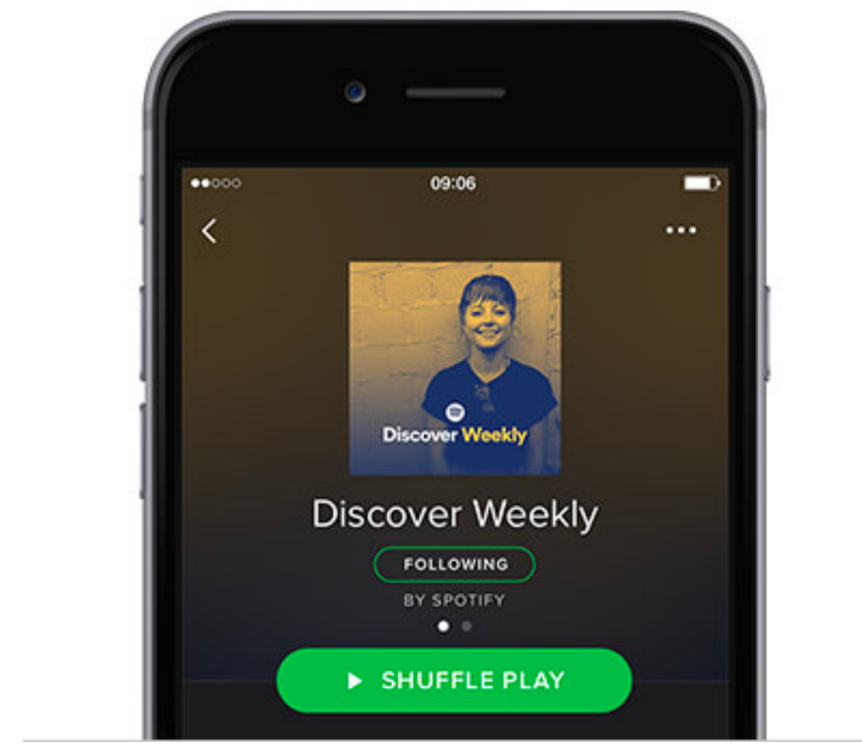
Music Industry

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™

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

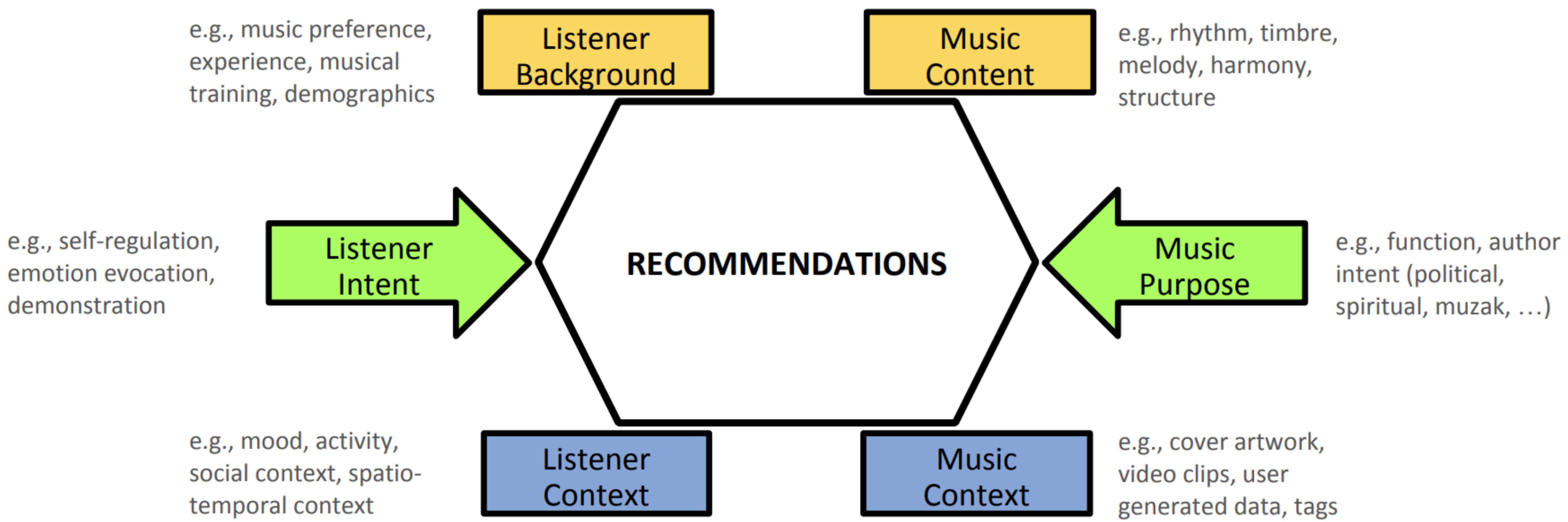
Repeated Consumption

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.)

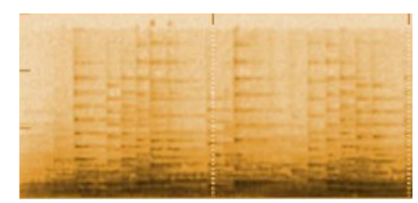
Music Recomendations

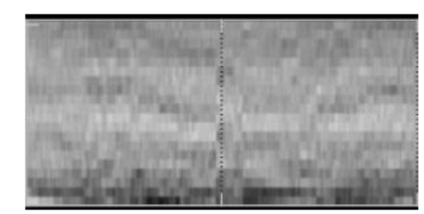
multiple factors: intrinsic (**personality** and **emotional state**) ,extrinsic (**activity**) and **contextual information** (weather, social surrounding, location,..) is needed to perform good recommendations

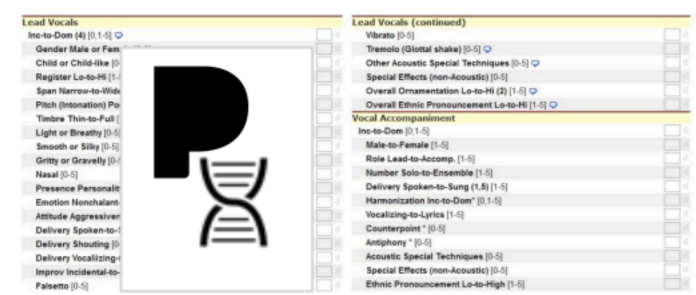

e.g., music preference, experience, musical training, demographics
Listener Background
Music Content
e.g., rhythm, timbre, melody, harmony, structure
e.g., self-regulation, emotion evocation, demonstration
Listener Intent
RECOMMENDATIONS
Music Purpose
e.g., function, author intent (political, spiritual,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

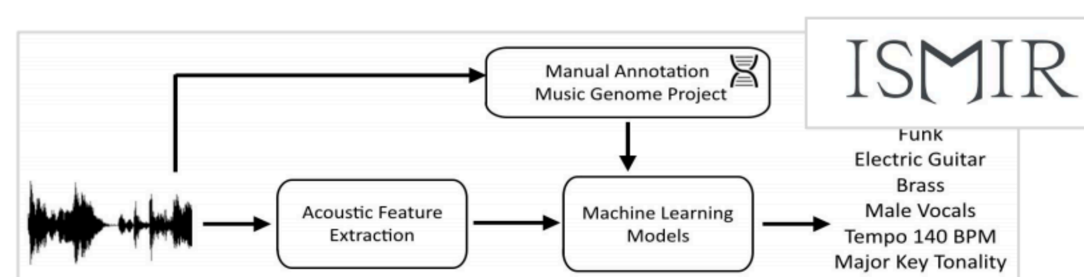
Data?

- Content (audio, symbolics, lyrics)
    - Machine listening/content analysis
    - Human labelling





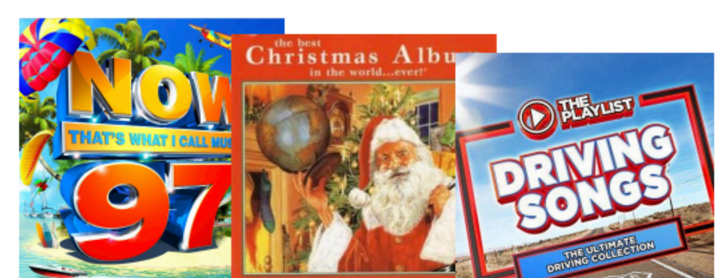




- Meta-data
    - Editorial
    - Curators
    - Multi-modal

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)

7digital MILLION SONG DATASET the echonest

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

Methods for recommending Music

pandora®

One of the best **content based** recommender music system

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 ca 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

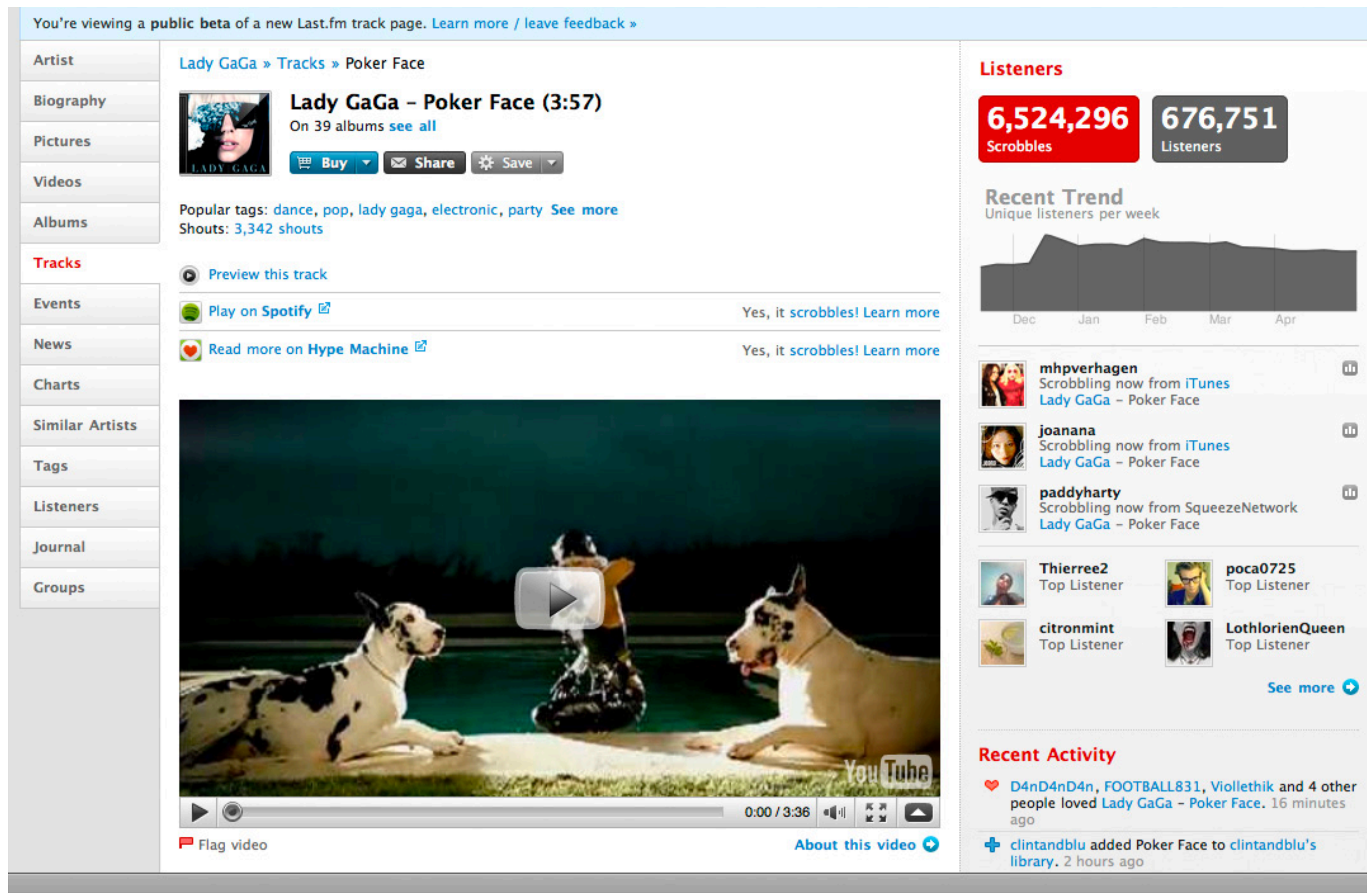
- Essentia (C++, Python): http://essentia.upf.edu/documentation/
- Librosa (Python): https://github.com/librosa
- Madmon (Python): https://github.com/CPJKU/madmom
- Marsyas (C++): http://marsyas.info/
- MIRToolbox (Matlab): https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/mirtoolbox
-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

Text Content Analysis

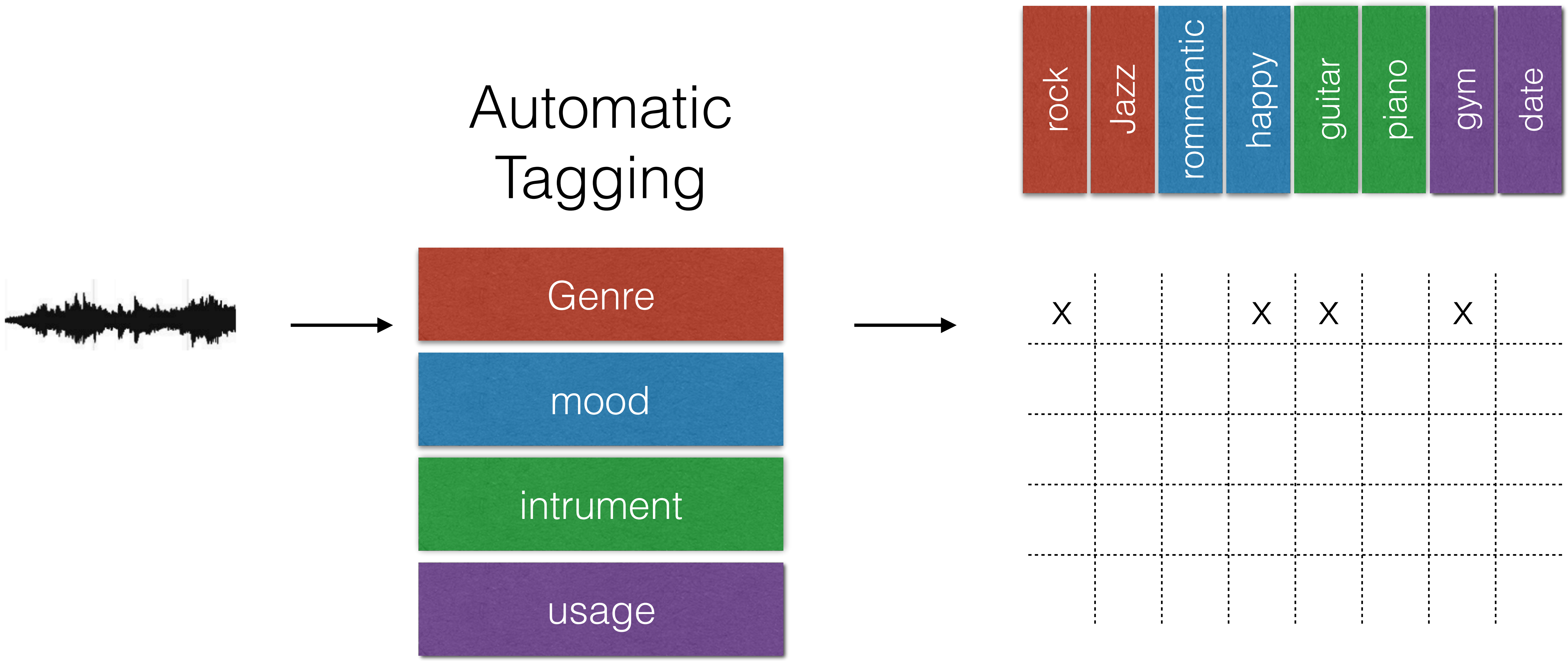
- Text processing of user-generated content and lyrics
  - Captures aspects beyond pure audio signals
  - no audio file is necessary
- Generate content features similarly as it is done in text files
  - Bag-of-Words, Vector Space, Tf-Idf
  - Topic Models, word2vect
- Sentiment analysis from reviews,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

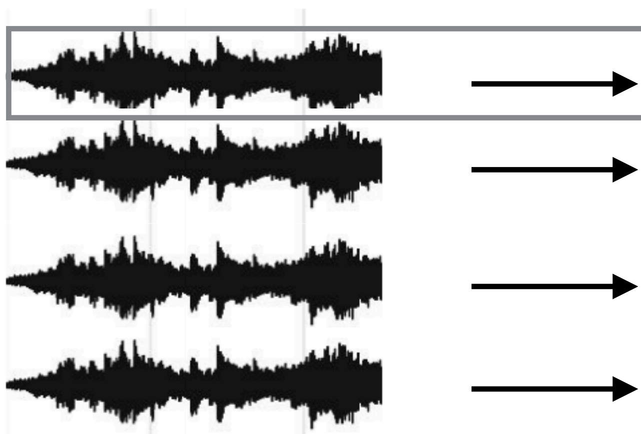


Content Based-Approach



Content Based-Approach

seed



	rock	Jazz	rommantic	happy	guitar	piano	gym	date
seed	X			X	X		X	
			X			X	X	
	X			X			X	
			X		X			X

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

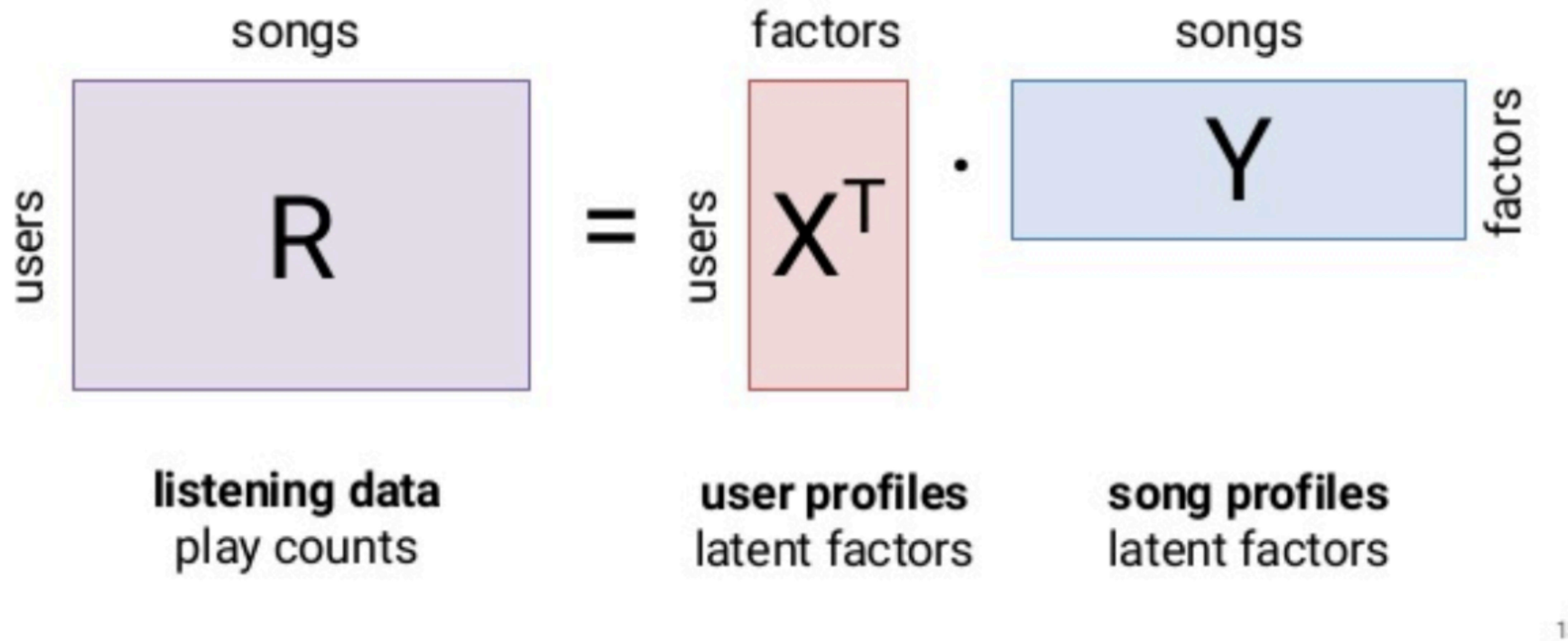
$$b_{ui} = \mu + b_{u,type(i)} + b_{u,session(i,u)} + b_i + b_{album(i)} + b_{artist(i)} + \frac{1}{|genres(i)|} \sum_{g \in genres(i)} b_g + c_i^T f(t_{ui})$$

Cold Start

- Collaborative Filtering methods fails on new item
- Solution:

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

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

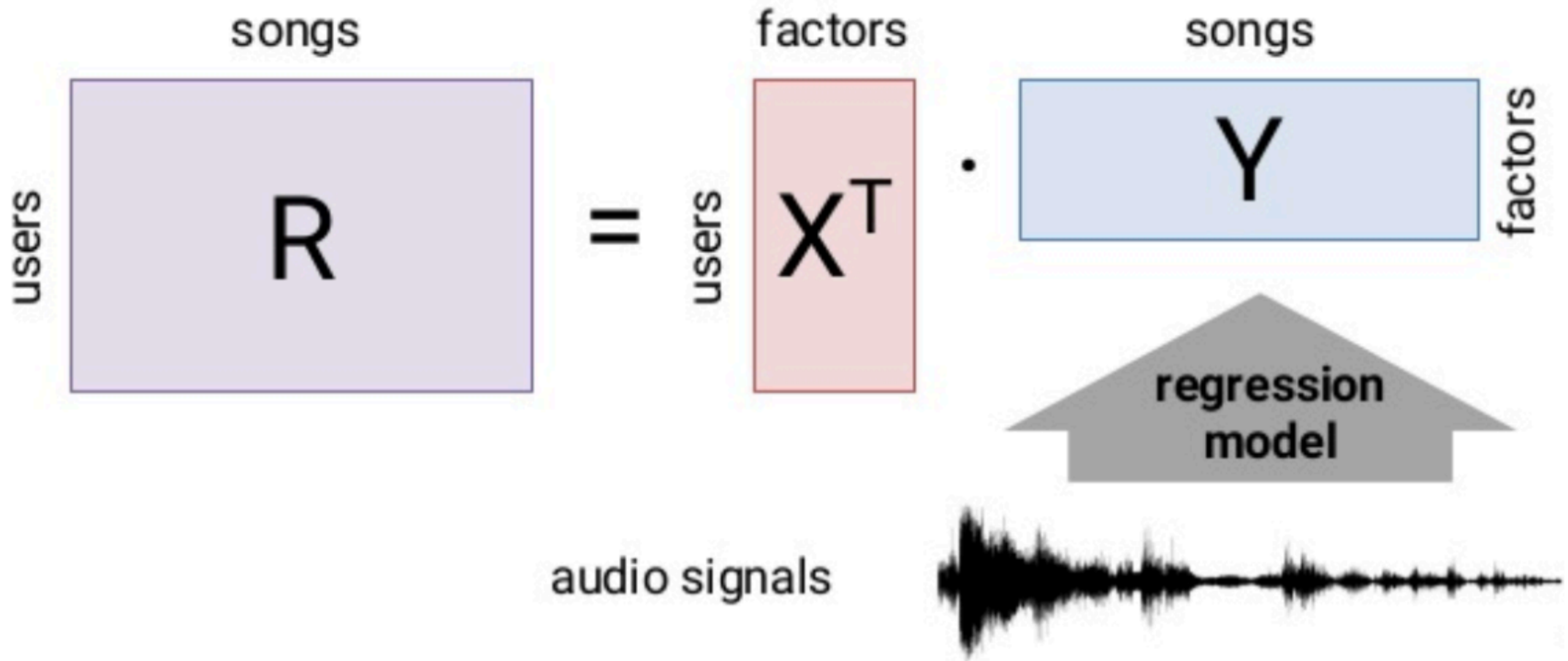


$$
\begin{aligned}
p_{ui} &= I(r_{ui} > 0), \\
c_{ui} &= 1 + \alpha \log(1 + \epsilon^{-1} r_{ui}).
\end{aligned}
$$

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

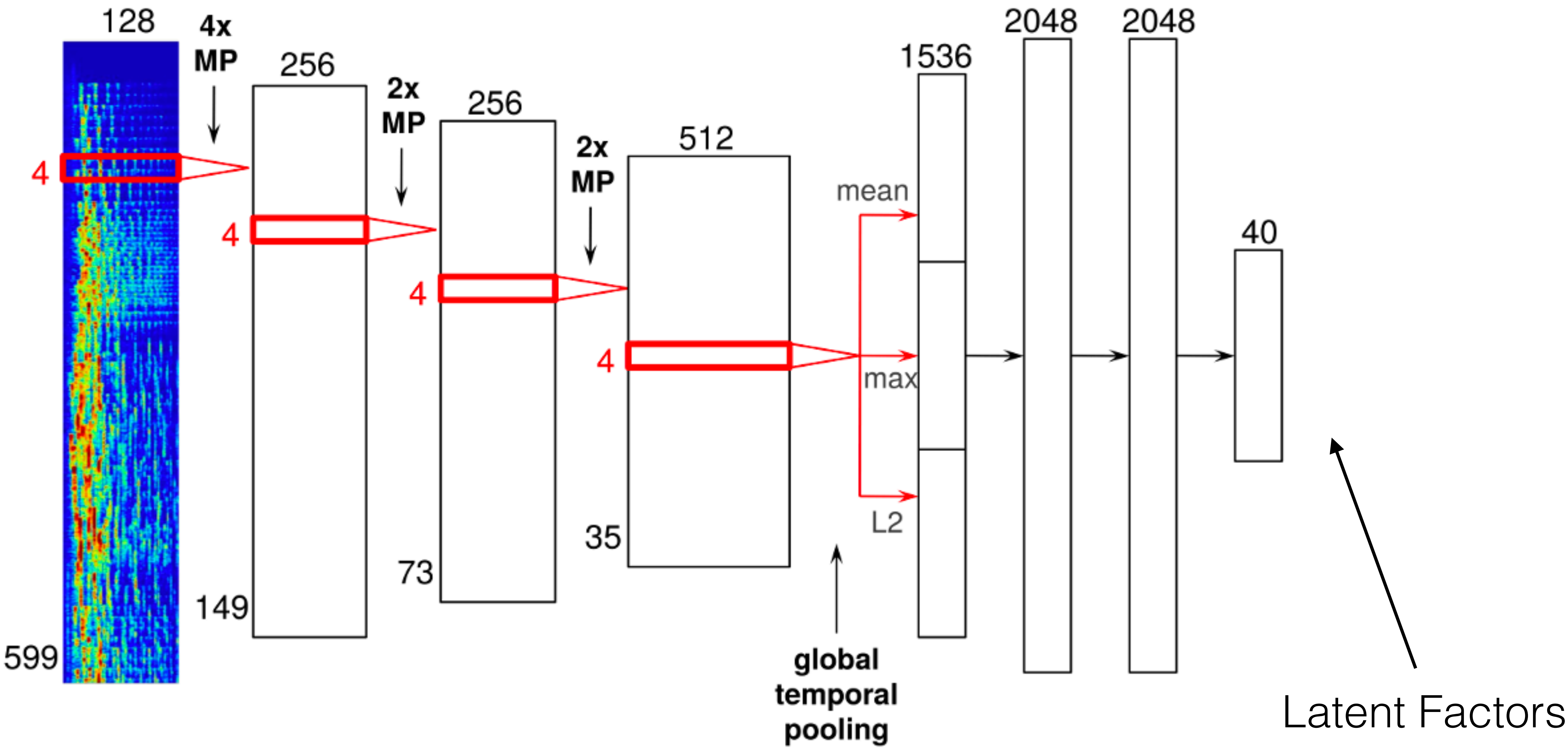
Deep content-based music recommendation A Van den Oord, S Dieleman, B Schrauwen Advances in neural information processing systems, 2643-2651	356	2013

Learn from audio the latent topics



UNIVERSITAT DE BARCELONA

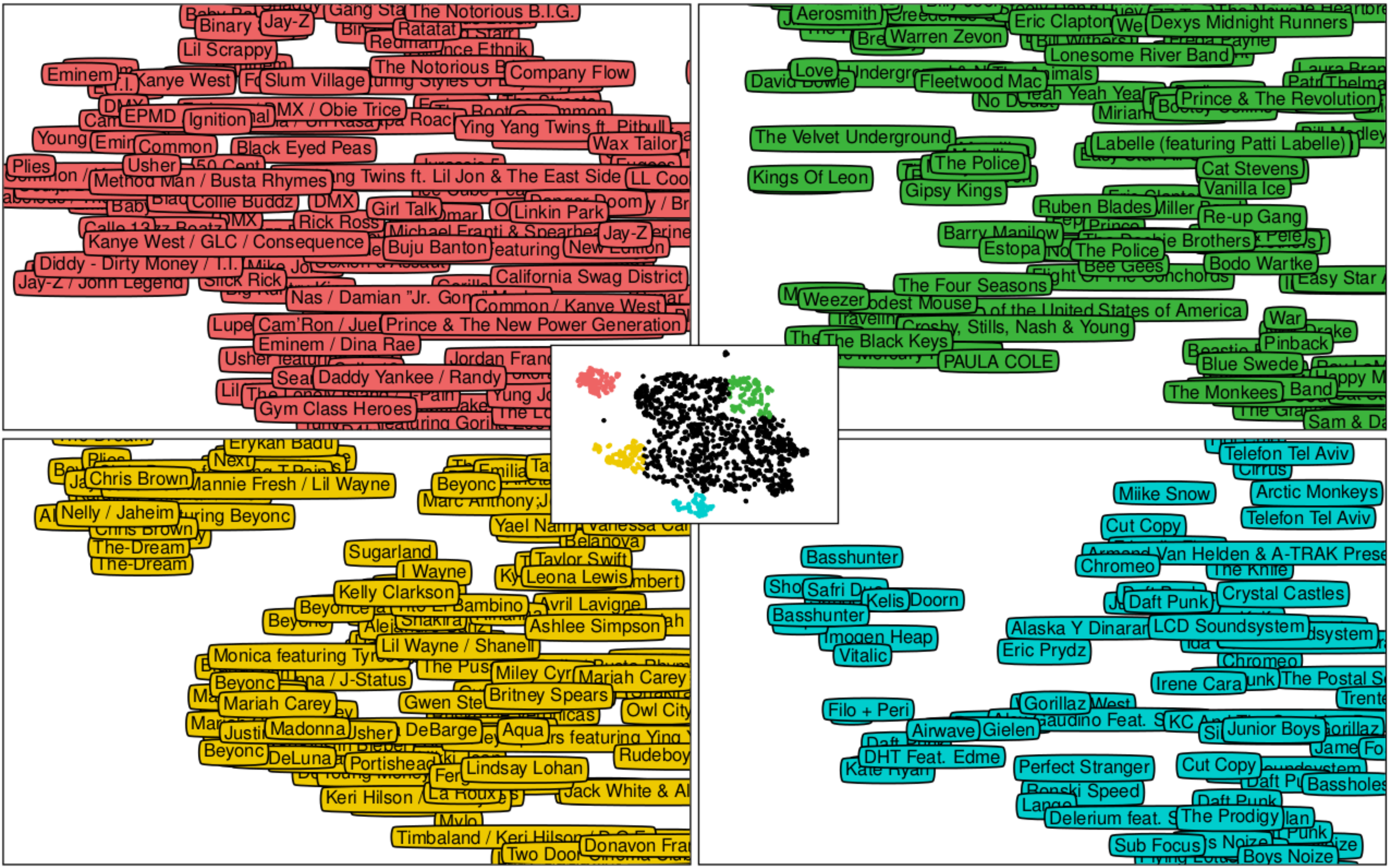
Raw audio models



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

356 2013

UNIVERSITAT DE BARCELONA



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

UNIVERSITAT DE BARCELONA

Temporal Aspect

- Recommending next tracks: **Temporal ordering matters!!**
- Notion of music rotation
  - Popularity catergories
  - 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

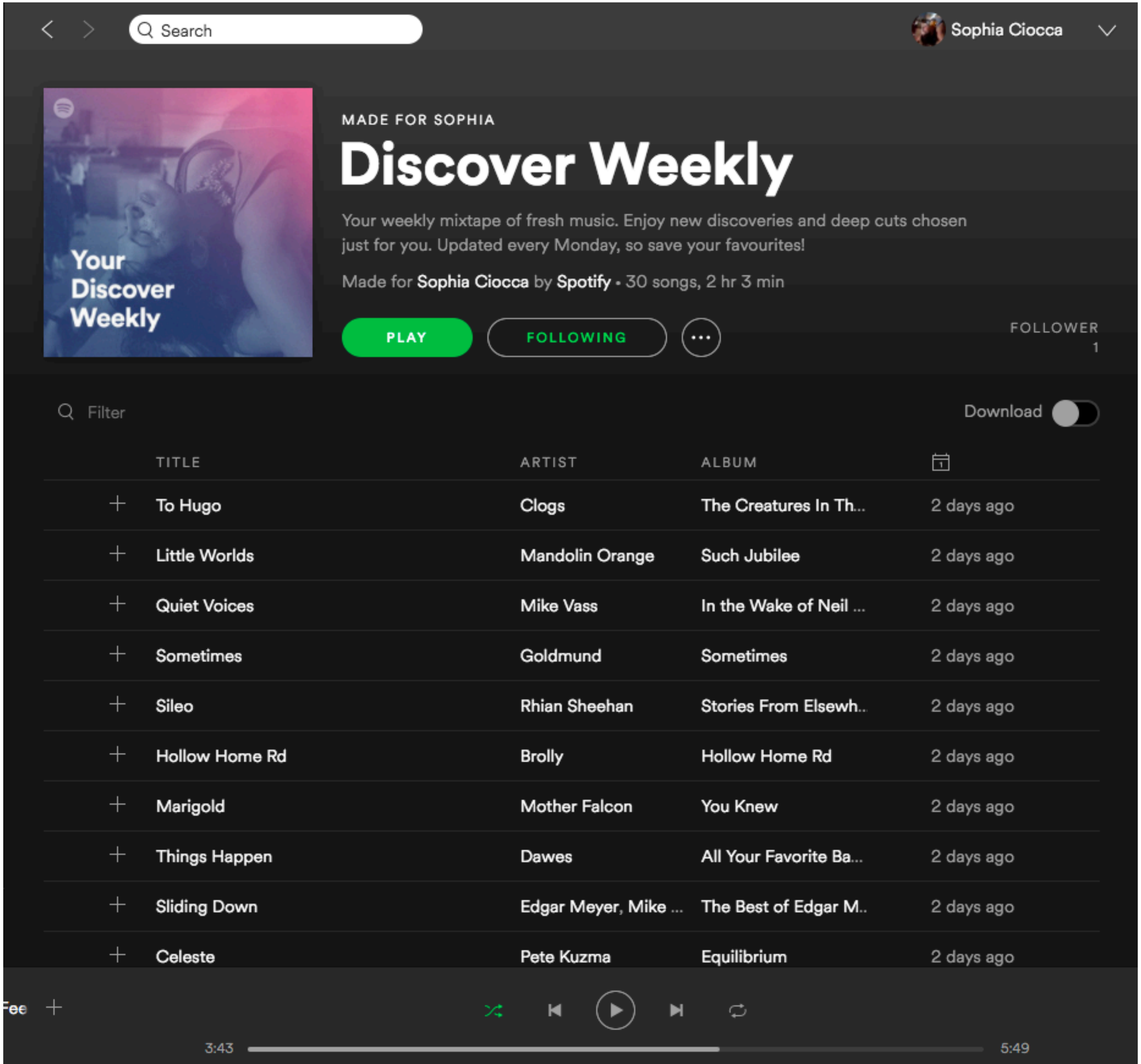
Accuracy (is not enough)

- 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



UNIVERSITAT DE BARCELONA

Spotify Architecture

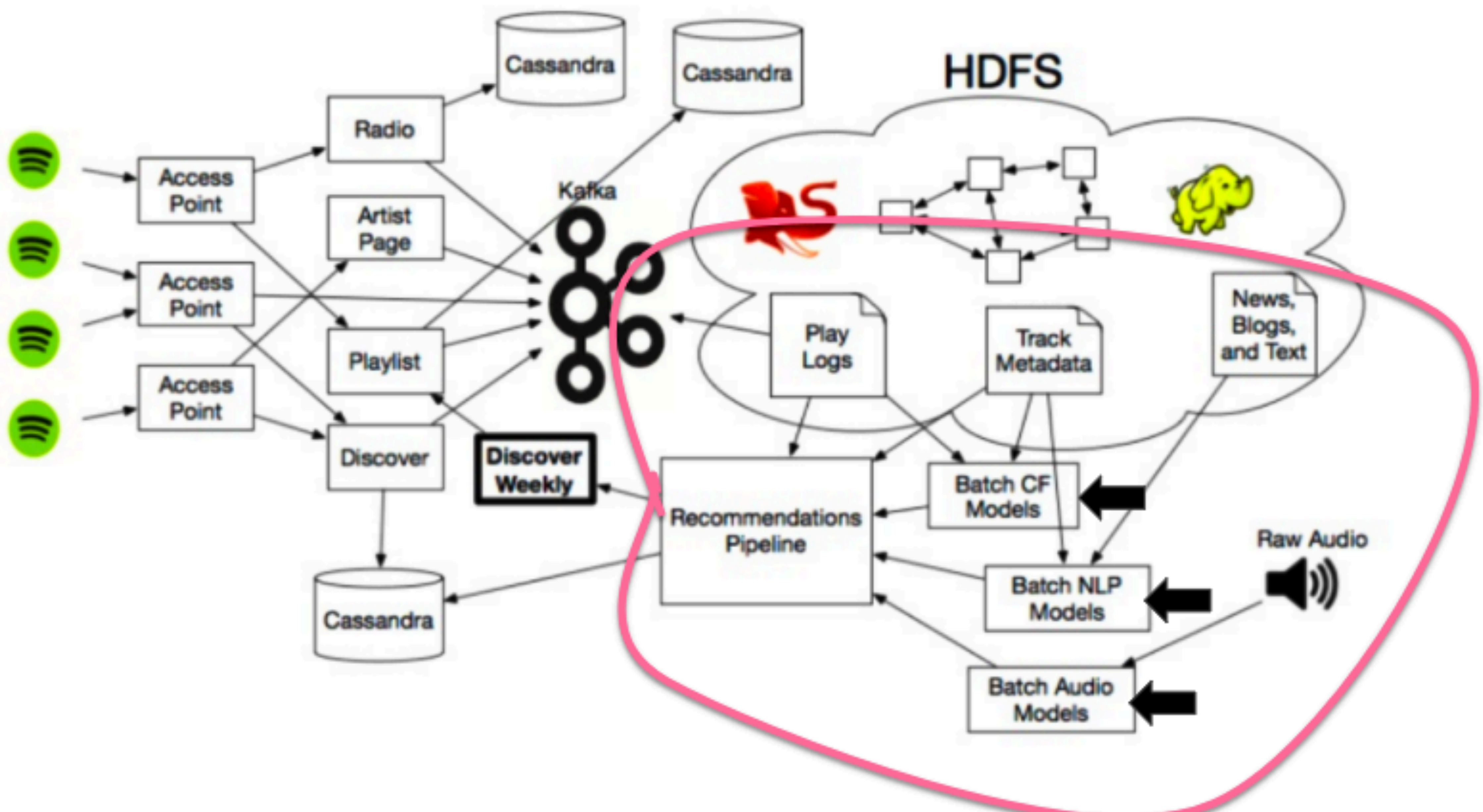


Image credit: Chris Johnson, Spotify

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.

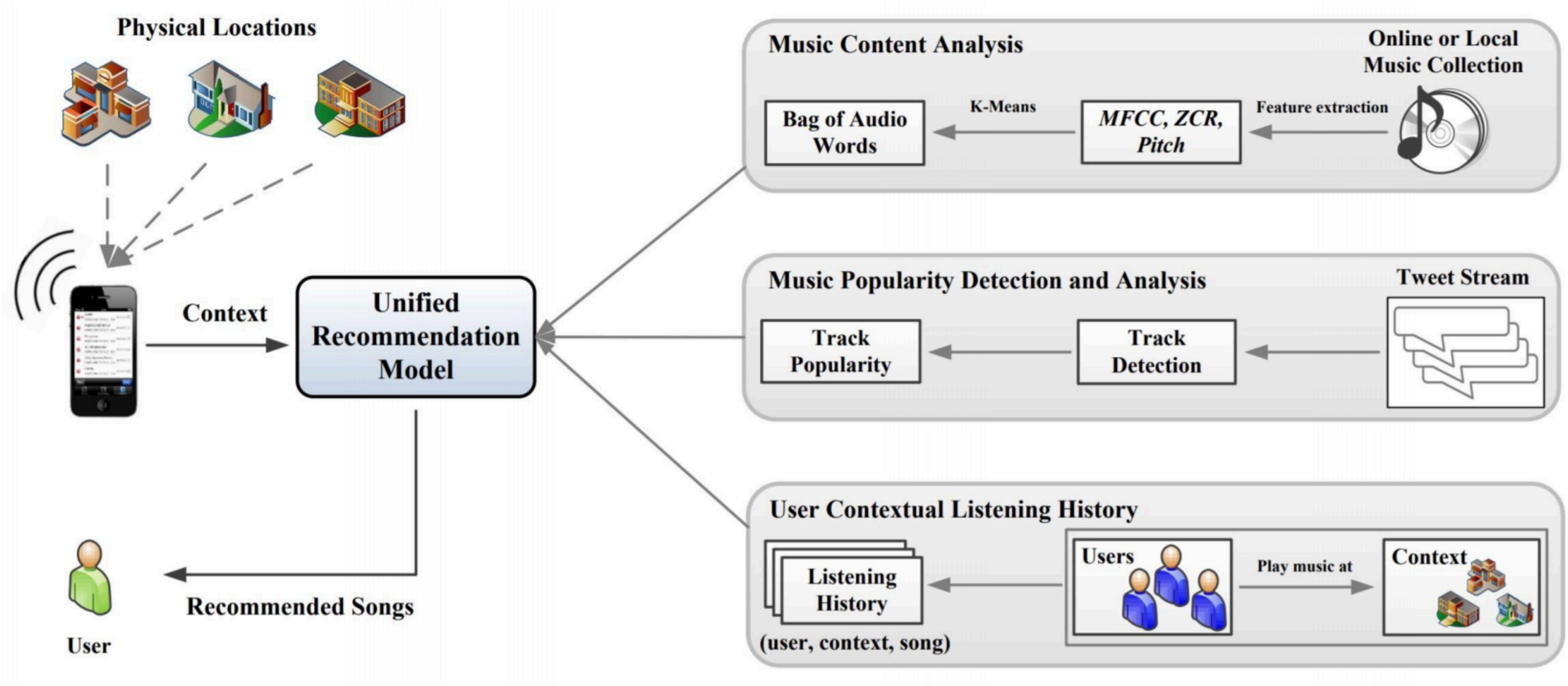
- **Each row represents one of Spotify's 140 million users** and **each column represents one of the 30 million songs** in Spotify's database.

CF in Spotify

- Matrix factorization using an optimization strategy

$$\min_{x,y} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i - \beta_u - \beta_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)$$

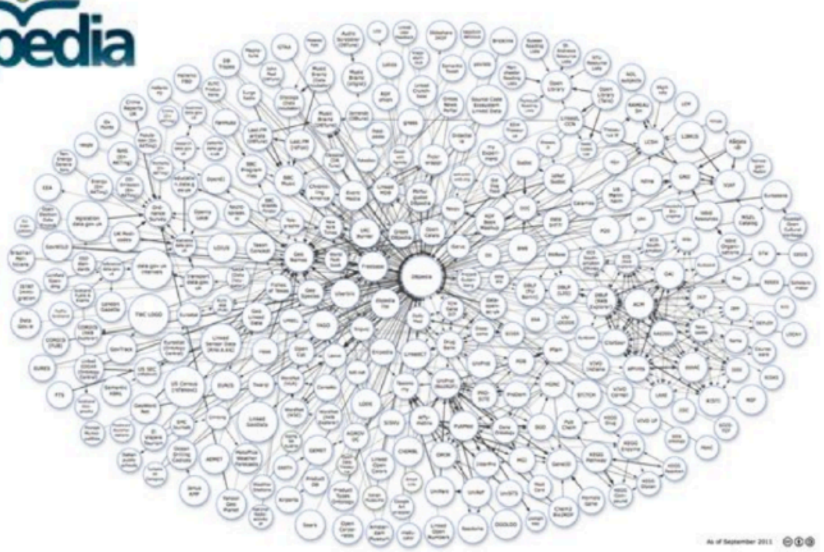
- When it finishes, we end up with two types of vectors, represented here by X and Y. **X is a *user* vector**, representing one single user's taste, and **Y is a *song*vector**, representing one single song's profile.



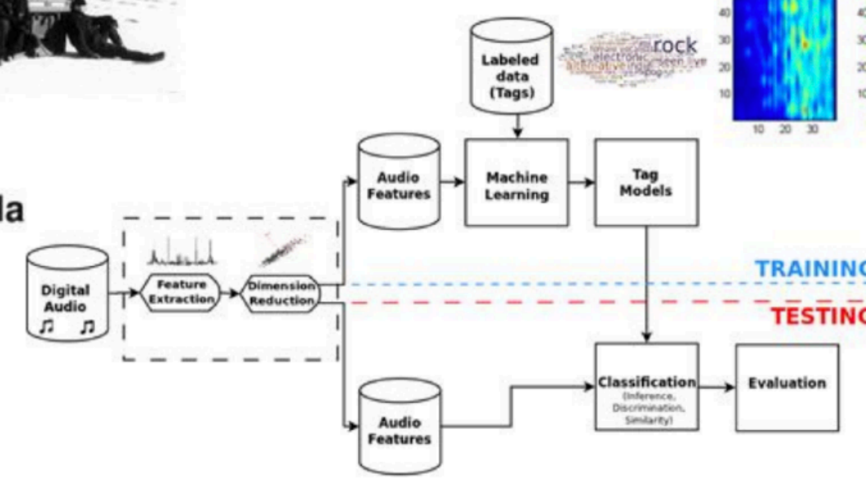
Just-for-me: An adaptive personalization system for location-aware social music recommendation | 52 | 2014
Z Cheng, J Shen
Proceedings of international conference on multimedia retrieval, 185

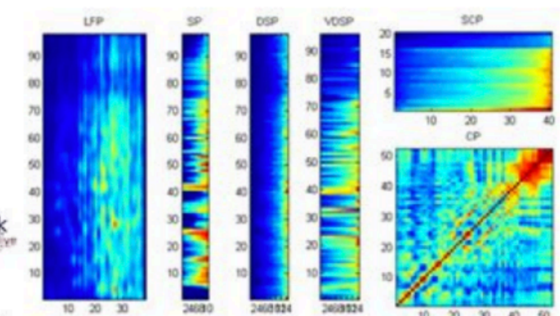
Hybrid MRS fusing knowledge-based recommendations and audio content-based recommendations obtained via auto-tagging (rank fusion)

DBpedia



+

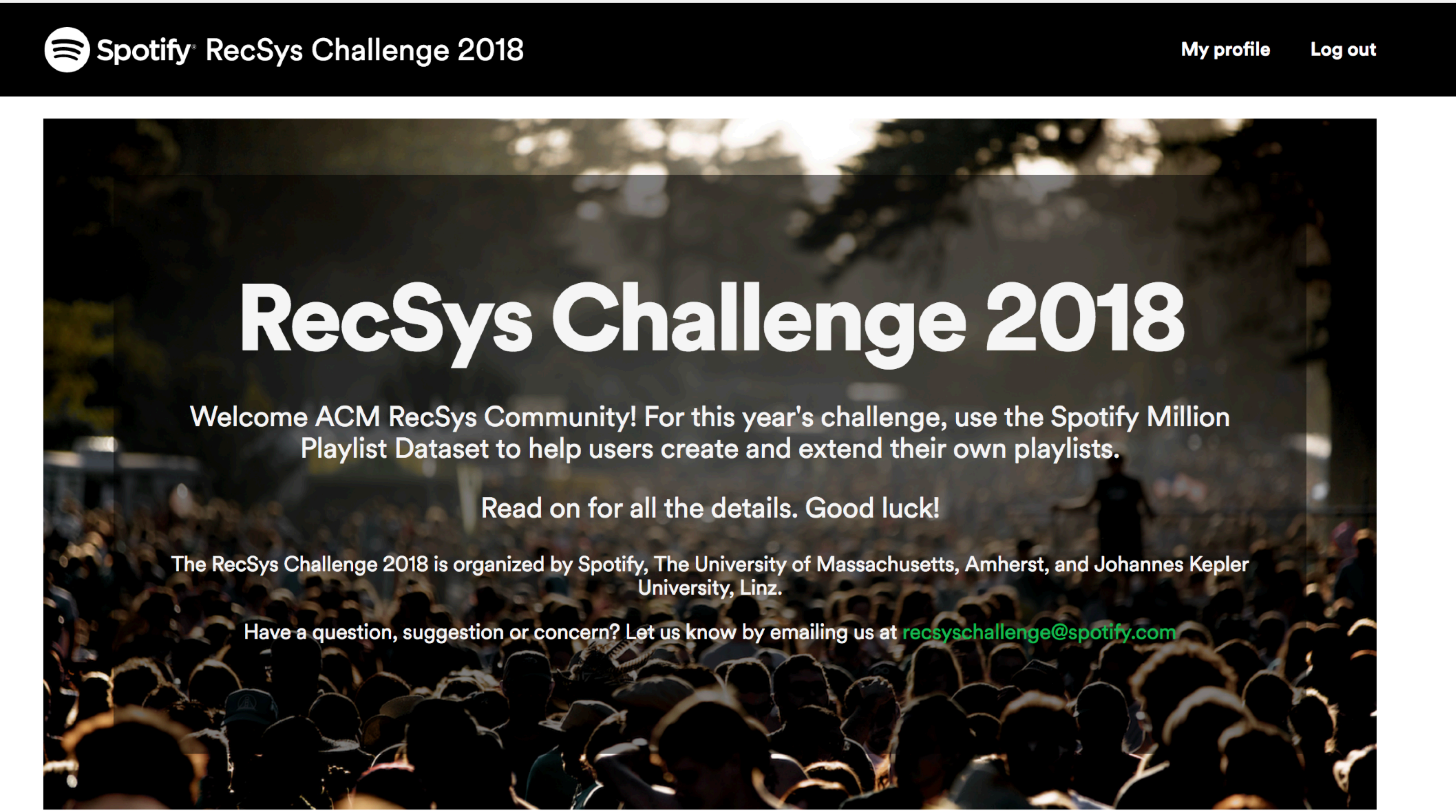




Location-aware music recommendation using auto-tagging and hybrid matching | 59 | 2013

M Kaminskas, F Ricci, M Schedl

Proceedings of the 7th ACM conference on Recommender systems, 17-24


Spotify RecSys Challenge 2018
My profile
Log out
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

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):

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

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:

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

UNIVERSITAT DE BARCELONA

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:

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

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.