Music Recommendation

SMC Summer School Course, May 28th 2019

Peter Knees

peter.knees@tuwien.ac.at
Outline

9:30 – 11:00  Music Recommendation – What is it about?
11:00 – 11:30  Coffee Break
11:30 – 13:00  Recommender Techniques and Algorithms
13:00 – 14:00  Lunch Break
14:00 – 15:30  Recommendation for Music Creators
15:30 – 16:00  Coffee Break
16:00 – 17:00  More Use Cases (incl. Group Work)
Sources

Music Similarity and Retrieval: An Introduction to Audio and Web-based Strategies

Recommender Systems Handbook (2nd ed.)
Chapter 13: Music Recommender Systems

Overview and New Challenges of Music Recommendation Research in 2018
Tutorial
by M. Schedl, P. Knees, F. Gouyon. ISMIR’18.
Intro
Music Consumption

GLOBAL RECORDED MUSIC INDUSTRY REVENUES 1999-2017 (US$ BILLIONS)

Source: IFPI

PERFORMANCE RIGHTS
Revenue from music reproduction:
- on AM/FM radio
- at public venues
(NB: Excluding perf. rights from Streaming)

PHYSICAL
e.g. CDs

STREAMING
- Internet radio & on-demand
- Ad-supported & subscriptions

GLOBAL RECORDED MUSIC REVENUES BY SEGMENT 2017
Music Recommendation

Discovery

RADIO

- Terrestrial radio
- Satellite radio

STREAMING

- Internet radio
- On-demand

Consumption

- Vinlys
- Cassette
- CDs
- Digital downloads

On-demand
Music Industry Changing Landscape

- Growing industry
- Accelerating transition: Physical → Streaming

Not just a format transition, but a fundamental revolution. Moving away from “Discover + Own” model, towards “Access” model

→ Change of paradigm: Recommending an experience, not just a product/item. Distributor now must guide listener in (never-ending) consumption, not just sell.
Influence of Tech Research

- "Access" can have different meanings
- New listening format still not well-defined... The field is wide open
- Lots of recent developments

- Lean-in vs. Lean-back
  - e.g. on-demand
  - e.g. Internet radio

<table>
<thead>
<tr>
<th>Format</th>
<th>Interactive</th>
<th>Non-interactive</th>
</tr>
</thead>
<tbody>
<tr>
<td>$€¥</td>
<td>Ad-supported / “free”</td>
<td>Subscription</td>
</tr>
</tbody>
</table>

→ High impact potential from tech. research

(NB: These are only examples, not an exhaustive mapping)
Looking at where $€¥ comes from is not the full picture…
… time spent listening, by media, tells a different story:

Revenue
(US, Source: RIAA, 2017)
- Other (including terrestrial radio)
- Streaming
- Physical
- Digital (excl. Streaming)

Time spent listening
(US, Source: Edison Research, 2017)
- Terrestrial radio
- Streaming
- Physical + Digital (excl. Streaming)
- Other
Music Discovery

- Streaming “taking over” physical & downloads
- But competing with terrestrial radio, too

The Quest for “Discovery”
Ongoing quest for defining listening format calls for:

- Innovative Discovery features
- Right balance between lean-in & lean-back experiences
Challenges in Building a Real-World Music Recommender
Automatic Playlists/Radio Stations

- Personalized radio stations, e.g.
  - Spotify radio
  - Apple Music
  - YouTube Music
  - Deezer
  - Pandora
  - Last.fm

- Continuously plays similar music
- Based on content and/or collaborative filtering
- Optionally, songs can be rated for improved personalization

[Image of a music application interface with the Radio section highlighted, showing options for recently played artists and stations.]
Automatic Radio Station Generation Problem

- A continuation problem
- Given a listener enjoying a particular musical experience (defined by the music itself, but also contextual factors and the listener’s intent), what recommendations can we make to extend this experience in the best possible way for the listener?
A “good” recommendation?

What makes a good recommendation:

- Accuracy
- Good balance of:
  - Novelty vs. familiarity / popularity
  - Diversity vs. similarity
- Transparency / Interpretability
- Listener Context

It’s about recommending a listening experience

Influential factors:
- Listener
- Musical anchor
- Focus / Intent


[Jannach, Adomavicius, 2016] *Recommendations with a Purpose*, RecSys

Accuracy (is not enough)

- Typically, recommendations are based on predicting the relevance of unseen items to users. Or on item ranking.
- For recommendations to be accurate, optimize to best predict general relevance
  - e.g. optimizing on historical data from all users
- Too much focus on accuracy $\rightarrow$ biases (i.e. popularity and similarity biases)
  - Tradeoff popularity vs. personalization (is pleasing both general user base and each individual even possible?...)
  - Particular risk of selection bias when RecSys is the oracle (e.g. station)
  - Single-metric Netflix Prize (RMSE) $\rightarrow$ only one side of the coin

Novelty

- Introducing novelty to balance against popularity (or familiarity) bias
- Both are key: Listeners want to hear what’s hype (or what they already know). But they also need their dose of novelty... Once in a while.
  - How far novel? (“correct” dose?)
  - How often?
  - When?, etc...

<table>
<thead>
<tr>
<th></th>
<th>“ Yep, novelty’s fine”</th>
<th>“No novelty, please!”</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Listener</strong></td>
<td>Jazz musician</td>
<td>My mother</td>
</tr>
<tr>
<td><strong>Musical anchor</strong></td>
<td>Exploring a new friend’s music library</td>
<td>Playlist for an official high-stake dinner</td>
</tr>
<tr>
<td><strong>Focus</strong></td>
<td>Discovery</td>
<td>Craving for my hyper-personalized stuff</td>
</tr>
</tbody>
</table>

“Yep, novelty’s fine”

“No novelty, please!”
Diversity

- Introducing diversity to balance against similarity bias
- Similarity ≈ accuracy
  - Trade-off accuracy vs. diversity
  - As for Novelty, adding Diversity is a useful means for personalizing and contextualizing recommendations

<table>
<thead>
<tr>
<th>Listener</th>
<th>“Yep, bring on diversity”</th>
<th>“No diversity, please!”</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (good) DJ</td>
<td></td>
<td>Exclusive Metal-head</td>
</tr>
<tr>
<td>Musical anchor</td>
<td>Station anchored on “90’s &amp; 00’s Hits”</td>
<td>Self-made playlist anchored on “Slayer”</td>
</tr>
<tr>
<td>Focus</td>
<td>Re-discovery, hyper-personalized</td>
<td>“Women in Post-Black Metal”</td>
</tr>
</tbody>
</table>

[Parambath, Usunier, Grandvalet, 2016] A Coverage-Based Approach to Recommendation Diversity on Similarity Graph, RecSys
Exploration vs. Exploitation

• Exploit:
  • **Data** tells us what works best now, let’s play exactly that
  • Play something **safe now**, don’t worry about the future
  • Lean-back experience
  • “Don’t play music I am not familiar with”

• Explore:
  • Let’s **learn** (i.e. gather some more data points on) what **might** work
  • Play something **risky now**, preparing for tomorrow
  • Lean-in experience
  • “I’m ready to open up. Just don’t play random stuff”

Exploration vs. Exploitation

Helps alleviate limited reach of some recsys:
- Coldplay, Drake, etc. vs. “Working-class” musicians (long-tail)
- Radio typically plays 10’s artists per week
- Streaming has the potential to play 100k’s artists per week
- Caveat of collaborative filtering-based algorithms

(CF-based recommendations, Last.fm data)

Transparency / Interpretability

“Why am I recommended this?”

If you like Bernard Herrmann

You might like “Gimme some more” by Busta Rhymes
Transparency / Interpretability

“Why am I recommended this?”

If you like Bernard Herrmann

You might like “Gimme some more” by Busta Rhymes

Because:
He sampled Herrmann’s work
Transparency / Interpretability

- Explain how the system works: transparency
- Increases users’ confidence in the system: trust
- Facilitates persuasion
- Fun factor → increases time spent listening
- Increases personalization (e.g. “because you like guitar”)
- Better experience overall
- Caveat: Users will then want to correct potentially erroneous assumptions → Extra level of interactivity needed


---

[Learn To Fly - Foo Fighters](https://example.com)

- Full Album
- There Is Nothing Left To Lose
  - Foo Fighters
  - 11 songs

[Features of this Track]
- hard rock roots, electric guitar wall-o-sound, extensive vamping, repetitive melodic phrasing, major key tonality, a vocal-centric aesthetic, electric guitar riffs, and many other similarities identified in the Music Genome Project.
Listener Context

- Special case of **explicit listener focus/intent**, e.g.:
  - Focus on newly released music (new stuff)
  - Focus on activity (e.g. workout)
  - Focus on discovery (new *for me*)
  - On re-discovery (throwback songs)
  - Hyper-personalized (extreme lean-back, *my best-of*)
  - etc.

→ Each specific focus defines:
  - Which recommendations are best?
  - Which **vehicle** for recommendations is best **(HOW to recommend)**?
Focus on: Discovering an artist

AutoPlay On
Keep the music playing with similar songs
Focus on: New music

Non-personalized vs. Personalized
Focus on: Re-discovery

Focus on stuff you know you like
Personalized, leaning towards exploit
Focus on: Hyper-personalized Discovery

About discovering new stuff. Intended to feel like it’s curated. Just. For. Me.

Leaning towards explore
Focus on: Lean-in experience

Lean in: Building Playlists

Music Recommendation

Summer School, May 28th 2019
Focus on: Mood / Activity

Non-personalized vs. Personalized
Recommender Systems
Recommender Systems

- Results of **digitization of all areas of life:**
  - Growing amounts of data artifacts available
  - User generated + commercial
  - Impossible to keep track/remain in charge of data

- Means to deal with these new opportunities by providing **tailored views onto data (personalization)**

- Provide right items (options, answers, …) at the right time

- Found in all areas, powers central services of digital economy
Recommenders are ubiquitous on the Web

Need new music?

Last.fm lets you effortlessly keep a record of what you listen to* from any player. Based on your taste, Last.fm recommends you more music and concerts!

*We have a special term for this, it’s called scroblbing.
What’s special to music recommendation?

- More and more relevant to the Music Industry with rise of streaming
- Wide range of duration of items (2+ vs. 90+ minutes),
  Lower commitment, items more “disposable”, low item cost
  → “bad” recommendations maybe not as severe
- Magnitude of available data items (Millions) & data points (Billions)
- Diversity of modalities (audio, user feedback, text, etc.)
- Various types of items to recommend (songs, albums, artists, audio samples, concerts, venues, fans, etc.)
- Recommendations relevant for various actors (listeners, producers, performers, etc.)
What’s special to music recommendation?

• Very often consumed in sequence
• Re-recommendation often appreciated (in contrast to e.g. movies)
• Often consumed passively (while working, background music, etc.)
• Yet, highly emotionally connoted (in contrast to products, e.g. home appliances)
• Different consumption locations/settings: static (e.g., via stereo at home) vs. variable (e.g., via headphones during exercise), alone vs. in group, etc.
• Listener intent and context are crucial
• Importance of social component
• Music often used for self-expression
Techniques and Algorithms
Data fuels recommenders

**Interaction Data**
- Listening logs, listening histories
- Feedback ("thumbs"), purchases

**User-generated**
- Tags, reviews, stories

**Curated collections**
- Playlists, radio channels
- CD album compilations
Data fuels recommenders

Content (audio, symbolic, lyrics)

• Machine listening/content analysis
• Human labelling

Meta-data

• Editorial
• Curatorial
• Multi-modal (album covers etc.)
Recommender Classification Scheme

Today’s focus

Collaborative Filtering (CF)

(based on users/community)

Content-based Recommenders *

(based on item’s content)

Knowledge-based Recommenders

(product finder)

Context-aware Recommenders

(based on the usage context)

Hybrid Recommenders

(a mixture of different approaches)

* NB: Here, content generally refers to an item’s properties, i.e. not necessarily descriptions derived directly from the contents of a digital representation of an item but also associated data/metadata. This is not a perfectly valid definition of content, but widely accepted in recommender systems.
Collaborative Filtering (CF)

- Exploits interaction data
- “People who listened to track A, also listened to track B”
- Main underlying assumption: users that had similar taste in the past, will have similar taste in the future
- Typical methods
  - Comparing rows/columns in matrix
  - Matrix factorization
Collaborative Filtering (CF) continued

- Different types of interaction data can be exploited:
  - implicit (e.g. plays, listening time)
  - explicit (e.g. thumbs, ratings)
- Task: completion of user-item matrix (matrix very sparse!)
- Stemming from “usage” of music → close to “what users want”
The User Item Interaction Matrix

$U = \{u_1, ..., u_n\}$ ... set of users,
$P = \{p_1, ..., p_m\}$ ... set of items,
$R$ matrix of size $n \times m$, cell $r_{i,j}$ corresponds to user $i$’s rating for item $j$

<table>
<thead>
<tr>
<th>Example</th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>3</td>
<td></td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User 4</td>
<td></td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>User a</td>
<td>5</td>
<td></td>
<td>3</td>
<td>4</td>
<td>?</td>
</tr>
</tbody>
</table>

Example task: predict missing rating (item 5) for active user $a$
User-Based CF Recommendation

Idea: identify similar users, use their ratings to predict missing rating

Algorithm outline:

1. Calculate similarity of active user to all users that have rated the item to predict
2. Select $k$ users that have highest similarity (*neighborhood*)
3. Compute prediction for item from a weighted combination of the item’s ratings of users in neighborhood (weights correspond to similarity)
User-Based CF Recommendation

1. Calculate similarity (=weight) of active user to all users that have rated the item to predict

- Commonly used for user similarity: *Pearson’s correlation*

\[
sim(a,u) = \frac{\sum_{p \in P'} (r_{a,p} - \bar{r}_a)(r_{u,p} - \bar{r}_u)}{\sqrt{\sum_{p \in P'} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P'} (r_{u,p} - \bar{r}_u)^2}}
\]

where \( P' \) is the set of items rated by both users and \( \bar{r}_u \) is the mean rating of user \( u \):

\[
\bar{r}_u = \frac{1}{|P'|} \sum_{p \in P'} r_{u,p}
\]

- Ranges from \(-1\) to \(+1\), requires variance in user ratings (else undefined), accounts for users’ rating biases (general high or low ratings) by subtracting mean rating
User-Based CF Recommendation

1. Calculate similarity (=weight) of active user to all users that have rated the item to predict

- Pearson’s correlation has shown to work best for this purpose
- Alternatives are (adjusted) cosine similarity (see later), Spearman rank correlation, Kendall’s τ correlation, mean squared differences, entropy, etc.

2. Select $k$ users that have highest similarity (neighborhood)

- Predefine $k$, sort according to similarity scores, and select $k$ highest (should not need any further explanation...)

Music Recommendation
User-Based CF Recommendation

3. Compute prediction for item from a weighted combination of the item’s ratings of users in neighborhood

- Predict rating $r'$ as weighted average of deviations from neighbors’ mean

$$r'_{a,p} = \bar{r}_a + \frac{\sum_{u \in K} \text{sim}(a,u) \times (r_{u,p} - \bar{r}_u)}{\sum_{u \in K} \text{sim}(a,u)}$$

- where $K$ is the set of the $k$ nearest neighbors and $\bar{r}_a$ the mean rating of the active user $a$ (this time calculated over all of $a$’s ratings)

- Starts from $a$’s rating bias and adds deviations based on similarity

- After predicting all missing values of $a$, the items with highest prediction will be recommended to $a$
User-Based CF Recommendation – Example

- Back to our example...
- User 2 hasn’t rated item 5...

1. Calculate correlations

\[
\text{sim}(a,u_i) = \frac{(5-4)(3-2.67) + (3-4)(2-2.67) + (4-4)(3-2.67)}{\sqrt{(5-4)^2 + (3-4)^2 + (4-4)^2}} \cdot \frac{\sqrt{(3-2.67)^2 + (2-2.67)^2 + (3-2.67)^2}}
\]

- \[
\text{sim}(a,u_1) = \frac{(5-4)(3-2.67) + (3-4)(2-2.67) + (4-4)(3-2.67)}{\sqrt{(5-4)^2 + (3-4)^2 + (4-4)^2}} \cdot \frac{\sqrt{(3-2.67)^2 + (2-2.67)^2 + (3-2.67)^2}} = \frac{0.33 + 0.67 + 0}{\sqrt{0.11 + 0.44 + 0.11}} = \frac{1}{1.15} = 0.87
\]
- \[
\text{sim}(a,u_3) = \frac{(5-4)(3-2.67) + (3-4)(1-2.67) + (4-4)(4-2.67)}{\sqrt{(5-4)^2 + (3-4)^2 + (4-4)^2}} \cdot \frac{\sqrt{(3-2.67)^2 + (1-2.67)^2 + (4-2.67)^2}} = \frac{0.33 + 1.67 + 0}{\sqrt{4.66}} = \frac{2}{3.05} = 0.65
\]
- \[
\text{sim}(a,u_4) = \frac{(3-3.5)(4-3.5) + (4-3.5)(3-3.5)}{\sqrt{(3-3.5)^2 + (4-3.5)^2}} \cdot \frac{\sqrt{(3-3.5)^2 + (4-3.5)^2}}{0.5} = \frac{-0.25 - 0.25}{0.5} = -1
\]

We will ignore all users that are negatively (or un-) correlated!
User-Based CF Recommendation – Example

2. Sort and select neighbors
   (for the setting $k=2$):
   i.e., $K = \{u_1, u_3\}$

3. Calculate the prediction for item 5 for user $a$

\[
r'_{a,i_5} = 4 + \frac{[0.87 \times (3 - 2.75)] + [0.65 \times (4 - 2.8)]}{0.87 + 0.65} = 4 + \frac{0.9975}{1.52} = 4.66
\]

- Thus, we predict a rating of 4.66 (or 4.5 or 5, depending on the scale)
- Is this a good prediction?
- What would be the predicted rating for item 2?
  And which of the two would you recommend to user $a$ → optional homework! :)

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>User 4</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>User $a$</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>?</td>
<td></td>
</tr>
</tbody>
</table>
Item-Based CF Recommendation

• Alternative approach (compare items/columns instead of users/rows)

• Better suited for large-scale recommenders than user-based CF

• Preprocessing can be performed offline, i.e., all *item-to-item similarities* can be calculated in advance (need update after some time)
  (Could be done for user-to-user similarities too, but...)

• \( n \) users and \( m \) items: in worst case \( n \times m \) evaluations

• More realistic: users rate only small number of items (\(<<m \) !)
  To predict item \( i \), find most similar (item-sim. matrix lookup), and weight own ratings over these items

• For item-based CF, at runtime, recommendation in real-time possible
  (e.g., Amazon used this [Linden et al., 2003])
Problems

Biggest problem for collaborative filtering:

**data sparsity!**

= most entries of the user-item rating matrix are empty

- Possibly millions of users and hundreds of thousands of users; but users just rate a few items; sparsity is the percentage of empty cells
- No overlap between user vectors or just based on a few items
- Correlation values become unreliable (e.g., consider the example of very high values based on two overlapping items that by chance are rated the same) → unreliable neighbor selection in user/item-based CF
- The more data is available, the better recommendations will be!
“Cold-Start” Problems

• “Cold-start” problems are a specific form of data sparsity (aka “ramp-up” problems)

• When new users or new items are introduced to the system

  • **new-user problem**: user has no or few ratings
    - problem for CF due to inability to compare to other users
    - problem also for content-based rec. because no user profile available
    - challenge to find items to rate first such that predictions improve (“preference elicitation”)

  • **new-item problem**: items has no or few ratings
    - problem for CF, no problem for “real” content-based rec.
    - issue also for obscure items, problem for non-mainstream users
    - “early-rater” or “first-rater” problem:
      no benefit for first people rating, can’t match to others;
      severe in news recommendation as new items come in constantly
Factors Hidden in the Data

Original assumption of first matrix factorization-based recommender systems:

- Observed ratings/data are interactions of 2 factors: users and items
- Latent factors are representation of users and items
Matrix Factorization (cf. SVD)

- Decompose rating matrix into user and item matrices of lower dimension $k$
- Learning factors from given ratings using stochastic gradient descent
  \[
  \min_{x_*, y_*} \sum_{r_{u,i} \text{ is known}} (r_{ui} - x_{ui}^T y_i)^2 + \lambda (\|x_u\|^2 + \|y_i\|^2)
  \]
- Prediction of rating: inner product of vectors of user $u$ and item $i$
- Factors not necessarily interpretable (just capture variance in data)

Latent Factor Examples from Movie Domain

Matrix Factorization for Music Recommendation

- For music, variants deal with specifics in data, e.g.,
- Learning factors and biases using hierarchies and relations in data
cf. [Koenigstein et al. 2011]

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


- Special treatment of implicit data (preference vs. confidence)

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

preference: \( p_{ui} = \begin{cases} 1 & r_{ui} > 0 \\ 0 & r_{ui} = 0 \end{cases} \)
confidence: \( c_{ui} = 1 + \alpha r_{ui} \)

[Hu et al., 2008] Collaborative Filtering for Implicit Feedback Datasets, ICDM.
Example of Collaborative Filtering Output

People who liked **Disturbed — The Sound of Silence**, also liked...

1. Bad Wolves — Zombie

2. Five Finger Death Punch — Bad Company

3. Disturbed — The Light

4. Metallica — Nothing Else Matters
Factors Hidden in the Data

Original assumption of first matrix factorization-based recommender systems:

• Observed ratings/data are interactions of 2 factors: users and items
• Latent factors are representation of users and items

• But it’s a bit more complex...
Factors Hidden in the Data

INTRINSIC
“What?”
- Listener Background
- Music Content

GOAL
“Why?”
- Listener Intent
- Music Purpose

CONTEXT
“Where & How?”
- Listener Context
- Music Context

INTERACTIONS [observed in data]
Factors Hidden in the Data

**INTRINSIC “What?”**
- Sound properties
  - e.g., rhythm, timbre, melody, harmony, structure

**GOAL “Why?”**
- Reason for composing
  - e.g., function, intent (political, spiritual, muzak, ...)

**CONTEXT “Where & How?”**
- Cultural embedding
  - e.g., cover artwork, video clips, reviews, user generated data, tags

**ITEMS**
- Music Content
- Music Purpose
- Music Context
Audio Content Analysis

• In contrast to e.g., movies: **true content-based recommendation!**

• Features can be extracted from any audio file
  → no other data or community necessary
  → no cultural biases (no popularity bias, no subjective ratings etc.)

• Learning of high-level semantic descriptors from low-level features via machine learning

• Deep learning now the thing
  (representation learning and temporal modeling directly from the signal,
  without hand-crafting features → CNNs, RNNs)


Audio Content Analysis: Selected Features

- **Beat/downbeat → Tempo:** 85 bpm
- **Timbre (→ MFCCs)**
  e.g. for genre classification, “more-of-this” recommendations
- **Tonal features (→ Pitch-class profiles)**
  e.g. for melody extraction, cover version identification
- **Semantic categories via machine learning:**
  not_danceable, gender_male, mood_not_happy

Different versions of this song:
- Simon & Garfunkel - The Sound of Silence
- Anni-Frid Lyngstad (ABBA) - En ton av tystnad etc.

Disturbed
The Sound of Silence
Audio Features: Basic Processing Steps

- Convert signal from time domain to *frequency domain*, e.g., using a Fast Fourier Transform (FFT)

- *Psychoacoustic transformation* (Mel-scale, Bark-scale, Cent-scale, ...): mimics human listening process (not linear, but logarithmic!), removes aspects not perceived by humans, emphasizes low frequencies

- Extract features
  - *Block-level* (large time windows, e.g., 6 sec)
  - *Frame-level* (short time windows, e.g., 25 ms)
    - needs model distribution of frames

- Calculate similarities between feature vectors/models
From Time to Frequency Domain (1 Frame)
Fourier Transform (FT) / Spectrogram

time domain

frames: sequence of samples

FT

spectrogram: visualization of signal in frequency domain
**Pitch Class Profiles** (aka *chroma vectors*)

- Transforming the frequency activations into well known musical system/representation/notation
- Mapping to the equal-tempered scale (each semitone equal to one twelfth of an octave)
- For each frame, get intensity of each of the 12 semitone (pitch) classes

(Fujishima; 1999)
Semitone Scale

- Map data to semitone scale to represent (western) music
- Frequency doubles for each octave
  - e.g. pitch of A3 is 220 Hz, A4 440 Hz
- Mapping, e.g., using triangular filter bank
  - centered on pitches
  - width given by neighboring pitches
  - normalized by area under filter

The note C in different octaves vs. frequency
Pitch Class Features

- Sum up activations that belong to the **same class of pitch** (e.g., all A, all C, all F#)

- Results in a 12-dimensional feature vector for each frame
- PCP feature vectors describe tonality
  - Robust to noise (including percussive sounds)
  - Independent of timbre (~ played instruments)
  - Independent of loudness
Pitch Class Profiles in Action
Mel Frequency Cepstral Coefficients (MFCCs) have their roots in speech recognition and are a way to represent the envelope of the power spectrum of an audio frame.

- The spectral envelope captures perceptually important information about the corresponding sound excerpt (*timbral aspects*).
- Sounds with similar spectral envelopes are generally perceived as “sounding similar.”

How can we characterize (parameterize) this curve?
The Mel Scale

- Perceptual scale of pitches judged by listeners to be equal in distance from one another

- Given Frequency $f$ in Hertz, the corresponding pitch in Mel can be computed by

$$m = 2595 \log_{10} \left( 1 + \frac{f}{700} \right)$$

- Normally around 40 bins equally spaced on the Mel scale are used
MFCCs

MFCCs are computed per frame

1. Framing
2. DFT: discrete Fourier transform on windowed signal
3. Mapping of spectrum to the Mel scale (melspectrogram, “melgram”), quantization (into e.g., 40 bins)
4. Logarithm of Mel-scaled amplitude (motivated by the way humans perceive loudness)
5. **perform Discrete Cosine Transform (DCT) to de-correlate the Mel-spectral vectors**
   - similar to FFT; only real-valued components
   - describes a sequence of finitely many data points as sum of cosine functions oscillating at different frequencies
     \[
     X_k = \sum_{n=0}^{N-1} x_n \cdot \cos \left( \frac{\pi}{N} \left( n + \frac{1}{2} \right) \cdot k \right) \quad k = 0, \ldots, N-1
     \]
   - results in \( n \) coefficients (e.g., \( n = 20 \))

\[\text{NB: performing (inverse) FT or similar on log representation of spectrum: “cepstrum” (anagram!)}\]
MFCC Examples

- Beethoven
- Shostakovich
- Black Sabbath
“Bag-of-frames” Modeling

- Full music piece is now a set of MFCC vectors
  - Variable amount of $n$-dim features vectors per piece ($n$… number of MFCCs)
  - Number of frames depends on length of piece
- Need summary/aggregation/modeling of this set
  - Average over all frames? Sum?
- Comparing two songs = comparing their feature distributions
- Implication: loss of temporal information
“Bag-of-frames” Modeling

- Practical solution: describe distribution of all these local features via statistics such as mean, var, cov
- “Quick-and-dirty” approach: compare these values directly
- Better: calculate distance of distributions, e.g. via Earth Mover’s Distance or Kullback-Leibler divergence
- For two distributions, $p(x)$ and $q(x)$, the KL divergence is defined as:

$$KL(p \parallel q) \equiv \int p(x) \log \frac{p(x)}{q(x)} \, dx$$

- Expectation of the log difference between the probability of data in one distribution ($p$) and the probability of data in another distribution ($q$)
MFCCs for Genre Classification

- For multivariate Gaussian distributions, a closed form of the KL-divergence exists

\[
KL_{(P||Q)} = \frac{1}{2} \left[ \log \frac{|\Sigma_P|}{|\Sigma_Q|} + Tr \left( \Sigma_P^{-1} \Sigma_Q \right) + (\mu_P - \mu_Q)^T \Sigma_P^{-1} (\mu_Q - \mu_P) - d \right]
\]

- \(\mu\) ... mean, \(\Sigma\) ... cov. mat., \(Tr\) ... trace, \(d\) ... dimensionality
- asymmetric, symmetrize by averaging: \(d_{KL}(P, Q) = \frac{1}{2} \left( KL_{(P||Q)} + KL_{(Q||P)} \right) \)
- not a metric!

- Use KL divergence on Gaussian model of MFCC “bag-of-frames” as kernel (gram matrix) for Support Vector Machines (SVMs) [Mandel and Ellis, 2005]
Alternative: Codebook Approach

1. Extract features (e.g., MFCCs from all frames) from all songs in training collection
2. Try to describe the resulting feature distribution/space by finding clusters → clustering step (e.g., k-means clustering)
3. Cluster centers are the codebook entries or “words” (cf. “bag-of-words”) → choice of $k$ defines the dimensionality of the new(!) feature vector space
4. For each song (new or in training set), find closest cluster center for each extracted frame feature vector and create histogram of how often each cluster center (word) is mapped
5. Normalize histogram
6. Histogram is $k$-dim global feature vector of song
7. Compare songs by comparing histogram feature vectors
Codebook Approach (2D Example)

Frame-wise features of
- ... of song 1
- ... of song 2
- △ ... of song 3

... cluster centers, k=4
Codebook Approach (2D Example)

counting “word” occurrences:

- ... [4, 7, 2, 3]
- ... [0, 3, 6, 4]
- △ ... [4, 7, 3, 4]

normalize:

- ... [0.25, 0.44, 0.13, 0.19]
- ... [0.00, 0.23, 0.46, 0.31]
- △ ... [0.22, 0.39, 0.17, 0.22]

= song feature vectors

vector space:

- simple similarity (Eucl., cos)
- efficient indexing
- ...
Limitations of “Bag-of-Frames”

- Loss of Temporal Information:
  - temporal ordering of the MFCC vectors is completely lost because of the distribution model (bag-of-frames)
  - possible approach: calculate delta-MFCCs to preserve difference between subsequent frames
- Hub Problem (“Always Similar Problem”)
  - depending on the used features and similarity measure, some songs will yield high similarities with many other songs without actually sounding similar (requires post-processing to prevent, e.g., recommendation for too many songs)
  - general problem in high-dimensional feature spaces!
A More General Approach

- Automatically learn the features from signal → deep learning architecture
- “End-to-End Learning”
- Input: spectrogram or Mel-spectrogram
- CNN architecture (or CRNN)
- Output: Single (e.g., genre) or multi-class labels (e.g., tags)
- Still: carefully design architecture of network
  - What is the task? (e.g., percussive vs harmonic or both)
  - Which properties are desired? (e.g. pitch invariances)
End-to-End Learning for Tags

- Automatic learning of audio features for tagging with CNN
- CNN properties:
  - translation, distortion, and locality invariance
  - musical features/events relevant to tags can appear at any time or frequency range

[Choi et al., 2016]
Architecture

- Input: 29.1 sec audio clips (MagnaTagATune clip length)
- 12 kHz downsampling, 256 samples hop size → 1,366 frames per clip
- Log amplitude Mel-spectrogram with 96 Mel bands
- ReLUs in conv. layers
- Batch normalization, dropout, ADAM optimization
- Output: 50 tags

<table>
<thead>
<tr>
<th>Mel-spectrogram</th>
<th>Conv 3×3×128</th>
<th>MP (2, 4) (output: 48×341×128)</th>
<th>Conv 3×3×384</th>
<th>MP (4, 5) (output: 24×85×384)</th>
<th>Conv 3×3×768</th>
<th>MP (3, 8) (output: 12×21×768)</th>
<th>Conv 3×3×2048</th>
<th>MP (4, 8) (output: 1×1×2048)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Output 50×1 (sigmoid)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
So, great ... why is this difficult then?

- “Objective” similarity measure
- Describes the output of the applied transformation
- Works well for genre and mood classification

- The resulting numbers represent a very narrow aspect of acoustic properties, describe no *musical* qualities (structure, development, time dependency, etc.)
- Which sound properties are important to whom and in which context?
- Lack of any personal preferences or experiences
- No consideration of multimodality of music perception
Mind the Semantic Gap!

- High-level: Musical concepts as perceived by humans
- Mid-level: High-level-informed combination of low-level features
- Low-level: Statistical descriptions of signal, machine-understandable data

- e.g. melody, themes, motifs + “semantic” categories: genre, time period, mood, etc.
- e.g. MFCCs, chroma + (latent) text topics *typically the level used when estimating similarity!*
- e.g. energy, zero-crossing-rate + text: TFIDF
Auto-Tagging

Learning semantic labels from content features

(Sordo; 2012)
Text Analysis Methods (Basic IR)

- Text-processing of user-generated content and lyrics
  → captures aspects beyond pure audio signal
  → no audio file necessary

- Transform the content similarity task into a text similarity task
  (cf. “content-based” movie recommendation)

- Allows to use the full armory of text IR methods, e.g.,
  - Bag-of-words, Vector Space Model, TFIDF
  - Topic models (LSI, LDA, …), word2vec

- Example applications: Tag-based similarity, sentiment analysis (e.g., on reviews), mood detection in lyrics

Using Texts for Music Recommendation

Recommending non-texts based on associated data, e.g., tags
Using Texts for Music Recommendation

Recommending non-texts based on associated data, e.g., web pages
Using Texts for Music Recommendation

Recommending non-texts based on associated data, e.g., reviews
Using Texts for Music Recommendation

Recommending non-texts based on associated data, e.g., tweets
Using Texts for Music Recommendation

Recommending music based on related texts, e.g., lyrics

Before day break there was none
And as it broke there was one
The Moon, the sun, it goes on 'n' on
The winter battle was won
The summer children were born
And so the story goes on 'n' on
Come woman if your life beats
Those we buried with the house keys
Smoke and feather where the fields are green
From here to eternity
Come woman on your own time

Around the world, around the world
Around the world, around the world
Around the world, around the world
Around the world, around the world
Around the world, around the world
Around the world, around the world
Around the world, around the world
Around the world, around the world
Around the world, around the world
Around the world, around the world

Describing Texts / Text-Based Features

- Extended meta-data is most frequently given as text (or could be transcribed to text), so we need to describe texts.
- Extract characteristics that allow description and algorithmic comparison ("features").
- Simple string comparison (character by character) is not very informative (and makes no sense).
- Need to extract the semantic content (topic) from the stream of characters (e.g., genre: sports vs. politics).
- Typically, the occurrence of specific words (terms) is a good indicator.
- Use descriptive statistics of word occurrences.
Describing Texts / Text-Based Features

- A **document** is a self-contained unit of text (including structural elements such as HTML or XML tags) which can be returned as a search result.

- A set of documents belonging together is referred to as corpus.

**Bag of Words (BoW)**

- Each document is represented as an unordered set of terms.

- Sequence of terms in a document is not considered important.

**Text Features: Vector Space Model (VSM)**

- Represent each document by a vector in a high-dimensional feature space (dimensionality = cardinality of term set).
- Typically, each dimension corresponds to the weight given to the respective term in the term set.
- Example: term set = [great, WoW, pop, concert, band, event, fantastic]

![BoW for document d](image)

<table>
<thead>
<tr>
<th>Term</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>great</td>
<td>0.12</td>
</tr>
<tr>
<td>wow</td>
<td>2.36</td>
</tr>
<tr>
<td>pop</td>
<td>0.46</td>
</tr>
<tr>
<td>concert</td>
<td>0.82</td>
</tr>
<tr>
<td>band</td>
<td>1.03</td>
</tr>
<tr>
<td>event</td>
<td>1.83</td>
</tr>
<tr>
<td>fantastic</td>
<td>1.42</td>
</tr>
</tbody>
</table>
Term weighting: monotonicity assumptions

1. **Rare terms are no less important than frequent ones (IDF assumption)**
   Importance of a term is the higher, the more rarely it appears among all documents (i.e. in the corpus)

2. **Multiple appearances of a term in a document are no less important than single appearances (TF assumption)**
   Importance of a term for a document \(d\) is the higher, the more often it appears in \(d\)

3. **Long documents are no more important than short documents**
   (normalization assumption)
   normalization by document/query length; usually performed in similarity computation
   (cosine measure) between \(q\) and \(d\)
Term weighting

- Weighting step crucial for VSM-based retrieval
- Assign a weight (an importance score) to each term $t$ in each document $d$
- How to compute the weight? → three monotonicity assumptions
  → $t$ is an important descriptor for $d$ if a token occurs frequently in the text and if it discriminates well between items
- Count how often each term $t$ appears in each document $d$ and in how many documents (over the whole collection)
  
  $tf_{d,t}$ ... term frequency  
  $df_t$ ... document frequency
- Assign a weight to each token for each document, frequently a variant of the tf·idf scheme ($idf$ ... inverse $df$, $m$ ... number of total items):

  \[ w_{t,d} = tf_{t,d} \cdot \log \frac{m}{df_t} \]
Text-Based Similarity Calculation

- Similarity calculation using the VSM:
- “Overlap score”: sum up over terms $i$ for which $a_i \neq 0$ && $b_i \neq 0$
- Euclidean distance
  \[
d(a,b) = \sqrt{\sum (a_i - b_i)^2}
\]
- L1 (Manhattan distance)
  \[
d(a,b) = \sum |a_i - b_i|
\]
- Cosine similarity
  preferred measure, document length has no influence on similarity!
- NB: many other similarity measures exist
Text-Based Features: Discussion

• Standard Information Retrieval approach can be applied to all domains if texts can be associated

• Text retrieval is well established but far from being perfect:
  • Tokenization eliminates the linguistic context, e.g., negations are modeled improperly (result: high VSM similarity between the phrases “no science-fiction movie” and “great science-fiction movie”)
  • VSM term vectors are usually very sparse: item-to-item similarity calculated in high dimensional space not reliable
  • Again, latent factor models might improve similarity calculation but not necessarily
Challenges for Context Methods

- Dependence on availability of sources (web pages, tags, playlists, ...)
- Popularity of artists may distort results
- Cold start problem (newly added entities do not have any information associated, e.g. user tags, users’ playing behavior)
- Hacking and vandalism (cf. last.fm tag “brutal death metal”)
- Bias towards specific user groups (e.g., young, Internet-prone, metal listeners on last.fm)
- (Reliable) data often only available on artist level for music context
- Content-based methods do not have these problems (but others)
Multimodal Approaches

- Incorporation of different sources / complementary information
- Content to handle cold-start problem in CF

- E.g. combining artist biography text embeddings with CNN-trained track audio embeddings


- E.g. fusing deep features from audio and image (album covers) and text

Feedback-Transformed Content

- CF model as target for learning features from audio
- Dealing with cold-start: predict CF data from audio
- Potentially: personalizing the mixture of content features

- E.g., learning item-based CF similarity function from audio features using metric learning

[McFee et al., 2012] Learning Content Similarity for Music Recommendation. IEEE TASLP 20(8).
Feedback-Transformed Content

- E.g. learning latent item features using weighted matrix factorization
  - CNN input: mel-spectrogram
  - CNN targets: latent item vectors
  - Visualization of clustering of learned song representations (t-SNE) on next slide

  [van den Oord et al., 2013] Deep Content-Based Music Recommendation. NIPS workshop.

- E.g. combining matrix factorization with tag-trained neural network to emphasize content in cold-start

  [Liang et al., 2015] Content-Aware Collaborative Music Recommendation Using Pre-Trained Neural Networks. ISMIR.
Feedback-Transformed Content

[van den Oord et al., 2013] Deep Content-Based Music Recommendation. NIPS workshop.
So much for the items

- Various ways to describe the items
- Recommendation hence completely detached from individual user/listener
- Not personalized: uses all of user data in one overall model

- Next: the user
Factors Hidden in the Data

**INTRINSIC**
"What?"
Listener Background
- Personal characteristics
  - e.g., music preference, experience, musical training, demographics

**GOAL**
"Why?"
Listener Intent
- Reason to select music/listen
  - e.g., self-regulation, emotion evocation, demonstration

**CONTEXT**
"Where & How?"
Listener Context
- Situation of listener
  - e.g., mood, activity, social context, spatio-temporal context
Listener Background

- Psychology- and sociology research driven area
- Goals: more predictive user models; dealing with user cold start
- Gathering information on user personality, music preference, demographics, cultural context, etc. (e.g., via questionnaires or predicted via other source)

Some findings: • age (taste becomes more stable);
  - when sad: open & agreeable persons want happy, introverts sad music;
  - individualist cultures show higher music diversity; etc.

[Ferwerda et al., 2016] Exploring music diversity needs across countries. UMAP.
Listener Context

- **Context categories and acquisition**: various dimensions of the user context, e.g., time, location, activity, weather, social context, personality, etc.

  Environment-related context
  - Exists irrespective of a particular user
  - Ex.: time, location, weather, traffic conditions, noise, light

  User-related context/background
  - Is connected to an individual user
  - Ex.: activity, emotion, personality, social and cultural context

Many more context categories

<table>
<thead>
<tr>
<th>Social context</th>
<th>Generic context</th>
<th>Technology context</th>
<th>Domain-specific context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social environment</td>
<td>- Cultural environment</td>
<td>- Task involvement / process</td>
<td>- Virtual environment</td>
</tr>
<tr>
<td>- Political circumstances and law</td>
<td>- Psychological predispositions and experiences</td>
<td>- Phase (e.g., start phase, final phase)</td>
<td>- Presence (e.g., virtual co-location, resource visibility)</td>
</tr>
<tr>
<td>- Micro-social environment</td>
<td>- Sociographics (e.g., sex, age)</td>
<td>- Degree of control / agency</td>
<td>- Interaction (e.g., coordination, communication)</td>
</tr>
<tr>
<td>- Organization</td>
<td>- Psychographics (e.g., personality, traits, affect, mood, attitude, emotions, experience, motivation)</td>
<td>- Obtrusiveness</td>
<td>- Discovery (e.g., service / resource discovery)</td>
</tr>
<tr>
<td>- Psychological predispositions and experiences (e.g., group dynamics, norms, social pressure, acceptance)</td>
<td>- Socioeconomics</td>
<td>- System behavior (e.g., system awareness, failure)</td>
<td>- Content (e.g., image, text, audio)</td>
</tr>
<tr>
<td>- Presence and behavior of people</td>
<td>- Perception</td>
<td>- System activity (e.g., pattern / speech recognition)</td>
<td>- Audiovisual (e.g., computer vision, visualization)</td>
</tr>
<tr>
<td>- Interaction with people</td>
<td>- Biophysical conditions (e.g., comfort, pain, physical fitness, heart rate)</td>
<td>- Efficiency and effectiveness (e.g., cost)</td>
<td></td>
</tr>
<tr>
<td>- Degree of formality (e.g., business / leisure environment, daily life, entertainment)</td>
<td>- Knowledge and skills (e.g., expertise, literacy, training, mental conditions, vocabulary, difficulty)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Habits (e.g., usage, browsing, recycling)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Degree of user profile stability</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Obtaining context data

- **Explicitly**: elicited by direct user interaction (questions, ratings in context)
  Ex.: asking for user’s mood or music preference (Likert-style ratings)

- **Implicitly**: no user interaction necessary
  Ex.: various sensor data in today’s smart devices (heart rate, accelerometer, air pressure, light intensity, environmental noise level, etc.)

- **Inferring** (using rules or ML techniques):
  Ex.: time, position → weather; device acceleration (x, y, z axes), change in position/movement speed → activity; skipping behavior → music preferences

Obtaining context data

Methods to establish relationship music context

1. Rating music in context
2. Mapping audio/content features to context attributes
3. Direct labeling of music with context attributes
4. Predicting an intermediate context

Putting it together

Catalog \( \xrightarrow{\text{match}} \) Data (e.g. audio content, interaction data, context, etc.) \( \xrightarrow{\text{Recommendation algorithm}} \) Recommendation
Putting it together

Catalog \(\textit{match}\) Data (e.g. audio content, interaction data, context, etc.)

Machine Learning e.g. Ensemble Learning

Recommendation

Feedback

Editorial/Curatorial
Content-Based
Collaborative Filtering
Context-based

Data (e.g. audio content, interaction data, context, etc.)

Catalog

Match
Recommendation Pipeline

Machine Learning
e.g. Ensemble Learning

Available music

Station

Playlist

Feedback

Adapt to specific focus/intent

Ranked list

Editorial/ Curatorial

Content- Based

Collaborative filtering

Personalized Filtering

Available music

Music Recommendation

SMC Summer School, May 28th 2019

113
Wait, what about time?

- Well… it’s important!

- “Music rotation rules” from AM/FM radio programming, e.g.:
  - Popularity categories: “Current”, “Recurrent”, “Gold”
  - Musical attributes: tempo, male vs. female vocals, danceability, major vs. minor, etc.
  - Sound attributes: synth vs. acoustic, intensity, etc.
  - Artist separation

[Price, 2015]: *After Zane Lowe: Five More Things Internet Radio Should Steal from Broadcast*, NewSlangMedia blog post
Several ways to consider time

• Predict best time for next user interaction with an item
  
  [Dai, Wang, Trivedi, Song, 2016]: Recurrent Coevolutionary Latent Feature Processes for Continuous-Time User-Item Interactions, Workshop on Deep Learning for Recommender Systems @ RecSys

• Modelling transitions in listening habits (e.g. artist transitions)
  
  [Figueiredo, Ribeiro, Almeida, Andrade, Faloutsos, 2016]: Mining Online Music Listening Trajectories, ISMIR

  [McFee, Lanckriet, 2012]: Hypergraph Models of Playlist Dialects, ISMIR

• Sequence-aware recommendation
  

  [Quadrana et al., 2018]: Sequence-Aware Recommendation, RecSys tutorial

Sequence-aware recommendation - Overview

Where does sequence-awareness fit?

Machine Learning  
e.g. Ensemble Learning

Available music

Strategy 1  Strategy 2  Strategy 3  ...  Strategy N

Current listening experience

Feedback

Adapt to specific focus/intent

Ranked list

Music Recommendation
Where does sequence-awareness fit?

Sequence learning

Current listening experience

Available music

Strategy 1

Strategy 2

Strategy 3

... Strategy N

Machine Learning

e.g. Ensemble Learning

Feedback

Adapt to specific focus/intent

Ranked list

Music Recommendation
Exploration vs. Exploitation

Content-Based Collaborative filtering Personalized Filtering Editorial/Curatorial

Learn balance between Explore vs. Exploit

Machine Learning e.g. Ensemble Learning

e.g. 80+ algorithms @

User Data
Open Research Challenges

- The missing parts!

- **Listener Intent**: Lots of insights from social psychology, cf. Laplante [2015], but less impact on actual music recommenders

- **Music Purpose**: somewhat less relevant, but still missing in the picture

- **Listener Background**: Gain deeper understanding of influence of emotion, culture, and personality on music preferences (also general vs. individual patterns)

---


[Knees, Schedl, Ferwerda, and Laplante, 2019 (expected)] *Listener Awareness in Music Recommender Systems*. Personalized Human-Computer Interaction, Augstein et al. (Eds.)
One more thing...

 USERS

 INTRINSIC
 "What?"
 Listener Background

 GOAL
 "Why?"
 Listener Intent

 CONTEXT
 "Where & How?"
 Listener Context

 ITEMS

 Music Content

 INTERACTIONS
 [observed in data]

 Music Purpose

 Music Context
Factoring the Service into the Picture

**INTRINSIC “What?”**
- **Listener Background**
- **Catalog**
- **Music Content**

**GOAL “Why?”**
- **Listener Intent**
- **Service Aims**
- **Music Purpose**

**CONTEXT “Where & How?”**
- **Listener Context**
- **Service Context**
- **Music Context**

**Music Recommendation**
Factors Hidden in the Data

**Users**

**Intrinsic**
- "What?"

**Goal**
- "Why?"

**Context**
- "Where & How?"

**Service**

**Catalog**
- Available music
  - e.g., licensed tracks, user provided content

**Service Aims**
- Intentions of service
  - e.g., “products” (discovery, etc.), promoting artists, maxing revenue

**Service Context**
- Operational circumstances
  - e.g., limitations (geo-, licensing restrictions)

**Items**

Music Recommendation
Looking into Service in More Detail

Recommendations (+collected data!) depend on factors other than users or items

- Which content is provided/recommended?
  - e.g. Soundcloud recommends different content than Spotify

- Why is this service in place? What is the purpose/identified market niche?
- What are the identified use cases? (Discovery? Radio? Exclusives? Quality?)
- Do they push their own content (cf. Netflix)?

- How do catalog and service aims depend on context?
- Are there licensing issues/restrictions in particular countries?
- Is the service context-aware? (e.g. app vs desktop/browser)
Maybe we need to talk about service biases

- Data from one service not generalizable to others
  
  ![SoundCloud](soundcloud.png) ≠ ![Bandcamp](bandcamp.png) ≠ ![Spotify](spotify.png) ≠ ![DEEZER](deezler.png) ≠ ![Pandora](pandora.png) ≠ …

- Particularly for niche market segments
  
  ![IDAGIO](idagio.png) ≠ ![Pono](pono.png) ≠ ![Qobuz](qobuz.png) ≠ …

- And different listening patterns (+content) in different parts of the world
  
  ![KKbox](kkbox.png) ≠ ![SuperPlayer](superplayer.png) ≠ ![Simfy Africa](simfy-africa.png) ≠ …

- Service influences listening behavior; it’s different to listening “in the wild”

- Focused service with clear customer base vs addressing all (market new products to underrepresented demographics)
Data Biases

- "Service biases" directly affect the data collected and therefore research datasets and experimentation
- Other biases in MIR datasets as well
  - Popularity biases (+feedback loops!)
  - Selection biases (no "alternate realities")
  - Cultural and community biases
  - Historical biases (symbolic, Classical music; licensing: royalty free)
- Impacts generalization of findings
The Bigger Picture
Example: Recommendation for Music Creators
You said “Music Industry Landscape”?

Music creation → Rest of Industry → Music Consumer

🎵 → $
Music Industry Landscape

Author
Composer
Songwriter

Performing Artist / Band
Engineer
Producer

Catalog/Rights Management
Music Publishing

Composition royalties
Sound recording royalties

Record Label
Aggregator

Hallelujah
Words & Music by Leonard Cohen

LET IT BE

Music Recommendation
Music Industry Landscape

PHYSICAL

Record Label

Physical manufacturer

Physical Retailer

Terrestrial Radio

Public performance

In Film and TV

In Ads

Advertisers

DIGITAL

Downloads Providers

Digital Streaming Services

Digital Radio Services

Social media Platforms

Music Consumer

Merchandising

Live Music Business

Artist Management

Digital Aggregator

Music Recommendation

SMC Summer School, May 28th 2019
Music Industry Landscape

Music creation → Rest of Industry → Music Consumer

Music creation
- Author
- Composer
- Singer/Songwriter
- Engineer
- Producer

Rest of Industry
- Catalog/Rights Management
- Physical manufacturer
- Physical retailer
- Record Label
- Terrestrial Radio
- Public performance
- In Film and TV
- In Ads
- Advertisers
- Digital Aggregator
- Downloads Providers
- Digital Streaming Services
- Digital Radio Services
- Social media Platforms

Music Consumer
- Merchandising
- Live Music Business
- Artist Management
Recommendation in the Creative Process
Factors Hidden in the Data … for Creators

INTRINSIC “What?”

Creator Background

GOAL “Why?”

Creator Intent

CONTEXT “Where & How?”

Creator Context

INTERACTIONS [observed in data]

ITEMS

Audio/Sound Content

A/S Purpose

Music Recommendation
Factors Hidden in the Data ... for Creators

**INTRINSIC**
“*What?”*

- Sound properties
  - e.g., timbre, texture, drum properties (ADSR)

**GOAL**
“*Why?”*

- Origin, Source of data
  - e.g., stylistic sample database (orchestra, vs. 8-bit, etc.)

**CONTEXT**
“*Where & How?”*

- Cultural embedding
  - e.g., usage by others/ references, tags

**ITEMS**

- Audio/Sound Content
- A/S Purpose
- Audio/Sound Context
Factors Hidden in the Data … for Creators

**INTRINSIC “What?”**
- **Creator Background**
  - Personal characteristics
    - e.g., music preference, experience, musical training

**GOAL “Why?”**
- **Creator Intent**
  - Reason to compose/produce
    - e.g., commissioned work, artistic expression

**CONTEXT “Where & How?”**
- **Creator Context**
  - Situation of creator
    - e.g., mood, activity, social context, spatio-temporal context
RecSys for Music Producers

• Today, basically all music and audio production becomes digital at one point
• Used tools reflect current practice of music making
  • Sound synthesis, virtual instruments, samples, pre-recorded material, loops, effects
  • Mixing, mastering, control for live performances
• Finding the right sound remains a central challenge:

  “Because we usually have to browse really huge libraries [...] that most of the time are not really well organized.” (TOK003)

  “Like, two hundred gigabytes of [samples]. I try to keep some kind of organization.” (TOK006)

• Actually the ideal target group for music retrieval and recommendation
Application: Tools for Music Creation

- Transcription
  - Analyze audio
  - Detect and classify instrument onsets
  - Generate symbolic representation
- Generation
  - Learn from symbolic representation
  - Pattern recognition and variation
- Live / Real-time
  - Follow performance and react
Digital Audio Workstations (DAWs)

- Commercial products come with very large databases of sounds
- Screen optimized for arrangement/mixing
- UI for finding material marginalized or external window
- Incorporated strategies:
  - Name string matching
  - Tag search/filtering
  - Browsing (=scrolling lists)
- Nobody tags their library!
Facilitating Sound Retrieval

- New (academic) interfaces for sample browsing

- Not so much recommendation. Why?
Let’s Ask the Users!

- Interviews, tests, and feedback sessions
  - Participatory workshops
  - Music Hack Days
  - Red Bull Music Academy
- Unique opportunity for research to get access to up-and-coming musicians from around the world
- Peer-conversations through semi-structured interviews
- Potentially using non-functional prototypes as conversation objects

[Andersen, Knees; 2016] *Conversations with Expert Users in Music Retrieval and Research Challenges for Creative MIR*. ISMIR.

The Role of Recommendation

- Recommenders are seen critical in creative work
  
  “I am happy for it to make suggestions, especially if I can ignore them” (TOK007)

- Who is in charge?
  
  “as long as it is not saying do this and do that.” (TOK009)

- Artistic originality in jeopardy
  
  “as soon as I feel, this is something you would suggest to this other guy as well, and then he might come up with the same melody, that feels not good to me. But if this engine kind of looked what I did so far in this track [...] as someone sitting next to me” (NIB4)

  “then it’s really like, you know, who is the composer of this?” (NIB3)

[Andersen, Grote; 2015] GiantSteps: Semi-structured conversations with musicians. CHI EA.
The Role of Recommendation (2)

- Users open to personalization, would accept cold-start

  “You could imagine that your computer gets used to you, it learns what you mean by grainy, because it could be different from what that guy means by grainy” (PA008)

- Imitation is not the goal: opposition is the challenge

  “I’d like it to do the opposite actually, because the point is to get a possibility, I mean I can already make it sound like me, it’s easy.” (TOK001)

  “Make it complex in a way that I appreciate, like I would be more interested in something that made me sound like the opposite of me, but within the boundaries of what I like, because that’s useful. Cause I can’t do that on my own, it’s like having a bandmate basically.” (TOK007)

Two recurring themes wrt. recommendation:

1. Virtual band mate (controlled “collaborator”)

   “I like to be completely in charge myself. I don’t like other humans sitting the chair, but I would like the machine to sit in the chair, as long as I get to decide when it gets out.” (TOK014)

2. Exploring non-similarity (“the other”, “the strange”)

   “So if I set it to 100% precise I want it to find exactly what I am searching for and probably I will not find anything, but maybe if I instruct him for 15% and I input a beat or a musical phrase and it searches my samples for that. That could be interesting.” (TOK003)

cf. defamiliarization: art technique to find inspiration by making things different
“The Other” in RecSys and Creative Work

• “Filter bubble” effects in recommender systems: obvious, predictable, redundant, uninspiring, disengaging results

• Responses: optimizing for diversity, novelty, serendipity, unexpectedness

• In particular in creative work
  • no interest in imitating existing ideas and “more of the same” recommendations
  • challenging and questioning expectations and past behavior

• For collaboration with an intelligent system for creativity, opposite goals matter:
  • change of context instead of contextual preservation
  • defamiliarization instead of predictability, explainability
  • opposition instead of imitation
  • obstruction instead of automation

[Zhao, Lee; 2016] How Much Novelty is Relevant?: It Depends on Your Curiosity. SIGIR.
Testing the Idea of Controlled “Strangeness”

• Instead of retrieving “more of the same” through top-N results

• As a response, we propose the idea of the Strangeness Dial

• Device to control the degree of otherness
  → turn to left: standard similarity-based recommendations,
  → turn to right: “the other”

• Built as a non-functional prototype (cardboard box) to enable conversations

• Also tested as a software prototype for strangeness in rhythm variation

[Knees, Andersen; 2017] Building Physical Props for Imagining Future Recommender Systems. IUI HUMANIZE.
Responses to the Strangeness Dial (Idea)

• Idea and concept are received well (via non-functional prototype)

"For search it would be amazing." (STRB006)

“In synth sounds, it’s very useful [...] Then the melody can also be still the same, but you can also just change the parameters within the synthesizer. That would be very cool.” (STRB003)

“That would be crazy and most importantly, it’s not the same strange every time you turn it on.” (TOK016)

• … but everybody understands it differently

“Strangeness of genre maybe, how different genre you want. [...] It depends how we chart the parameter of your strangeness, if it’s timbre or rhythm or speed or loudness, whatever.” (STRB001)

“No, it should be strange in that way, and then continue on in a different direction. That’s the thing about strange, that there’s so many variations of strange. There’s the small, there’s the big, there’s the left, there’s the right, up and down.” (STRB006)
Responses to the Strangeness Dial (Prototype)

- The software prototype tried to present “otherness” in terms of rhythm
- This was perceived by some but didn’t meet expectations of the majority
  - “I have no idea! It's just weird for me!” (UI03)
  - “It can be either super good or super bad.” (UI09)
- Concept is highly subjective, semantics differ
- Demands for personalization (i.e., “which kind of strange are you talking about?”)
  - “Then you have a lot of possibility of strange to chose from, actually. Like for me, I would be super interested to see it in ‘your’ strange, for example.” (STRB006)
Some Takeaways

- User intent is a major factor
- Experts need recommenders mostly for inspiration: serendipity is key
- Control over recommendation desired (...transparency could help)
- Not much collaborative interaction data in this domain
  → Strong focus on content-based recommenders
  → To find what is unexpected, new sources of (collaborative) usage data need to be tapped
- Making music is mostly a collaborative task and a useful recommender needs to be a collaborator
Trending Topics

- Intelligent machines to support music creation
- Many **supportive system prototypes and tools** in products, e.g.,
  - melody/composition: Lumanote, JamSketch
  - rhythm: Vogl [2017], Reactable STEPS/SNAP
  - “semantic” control, automatic remixes, …

- **AI for automatic composition**
  - Generative models
  - Producing royalty-free music (?)

[Granger et al., 2018] *Lumanote: A Real-Time Interactive Music Composition Assistant.* MILC@IUI.
[Cartwright, Pardo, 2013] *Social-Eq: Crowdsourcing An Equalization Descriptor Map.* ISMIR.
[Davies et al. 2014] *AutoMashUpper: automatic creation of multi-song music mashups.* TASLP.
AI-based Music Generation

Google Magenta
- deep neural networks for, e.g., expressive renderings, interpolations

Flow Machines/Spotify
- automatic continuation/accompaniment, composition in style of X

Jukedeck, melodrive, et al.
- Automatic, royalty-free soundtracks, video game music, “personalized music”

Other big tech companies somewhat active as well: IBM Watson (Beat), Baidu

Further sources on generative music:
- How Generative Music Works: A Perspective (https://teropa.info/loop/)
- Neural Nets for Generating Music (Medium)
Working with Watson

Grammy award-winning music producer Alex Da Kid paired up with Watson to see if they could create a song together. Watson’s ability to turn millions of unstructured data points into emotional insights would help create a new kind of music that for the first time ever, listened to the audience.

Cognitive creation

Alex Da Kid used Watson’s emotional insights to develop ‘heartbreak’ as the concept for his first song, ‘Not Easy,’ and explored musical expressions of heartbreak by working with Watson Beat. Alex then collaborated with X Ambassadors to write the song’s foundation, and lastly added genre-crossing artists Elle King and Wiz Khalifa to bring their own personal touches to the track. The result was an audience-driven song launching us all into the future of music.

RecSys just an intermediary step to personalized content creation?
Where could this be going?

- Parameters of music + usage patterns, context, etc.
  \[\rightarrow\text{ train generative model to generate “the right music” for free?}\]
- Does music need to be “good” to be a success, i.e., listened to?
- (in AI terms: will the Turing test be passed?)
- In any case: music production will get increasingly automatized
Wrapping up + Outlook
Further use cases

- Alternative audio content to music, e.g.
  - Ads (where a lot of $$$ is)
  - News, Podcasts
  - Artist messages

- Central battle-place of competition with AM/FM radio
  - Streaming in a better place for ads-targetting
  - Radio in a better place for alternative content

- Open problems:
  - How to sequence different types of content? (i.e. what content when?)
  - How to personalize?
  - How to present it to the listener?
  - How to blend music and audio in social media platform experiences?
Further use cases

- Live Music Business, e.g.
  - Recommending upcoming concerts to listeners
  - Recommending artists to e.g. music festivals

- Recommendations for artist management, e.g.
  - Help agents find best opportunities for artists

- Recommendations to artists
  - Recommending artists where to play
  - Help artists grow their careers, with insights based on data
  - Help artists communication with their fanbase
Further use cases

- Data Science for record labels, e.g.
  - Assist A&R in finding new talents
  - An artist is launching an album, which track(s) to promote?
  - Make the best use / better monetization of back-catalogue
  - General assistance in business decisions
  - Marketing (where, to whom, how)
  - etc.

NB: Interesting explore/exploit trade-off
Further opportunities

• Voice-driven interaction with music
• Dedicated hardware (for home or car) vs. usual interfaces (e.g. phone)
• Smart speaker growth
• Today: “command-and-fetch”, e.g. “Play God’s Plan by Drake”
• Tomorrow: More casual interactions, ambiguous queries, conversations
• Calls for: Metadata, Personalization
• Competes with terrestrial radio (more passive listening)

[Dredge; 2018] Everybody’s talkin’: Smart speakers and their impact on music consumption, Music Ally Report fo BPI and ERA.
Ethics

- Business-related recommendations (e.g. promotional content) vs. what the user actually wants/needs
- Impact on popular culture (shaping what makes popular culture)
  - Responsibility to counteract algorithmic biases and business-only metrics
  - “Filter bubble”
- Impact on accessibility
e.g., are we all equal in the eyes of (ASR) technology?
- Impact on “how” people listen to music (e.g. influence on curiosity)
- Impact on artists, on what’s successful, on the type of music composed
- Privacy

Challenges

• Recommending diverse types of content
• Understanding listening behavior in context
• Blending social interactions in music streaming
• Blending human-curated recommendations with algorithmic ones
• Transparency and trust
• Managing a listener’s plurality of tastes without being disruptive
• Metrics for approximating long-term user satisfaction
• Voice-driven music interactions (in car, at home)

Take-Away Messages

• Dramatic changes in music consumption (growth, ownership → access) imply great challenges and impact for recommender systems

• Music is not “just another item”, many different representations and sources of data for manifold recommendation techniques

• Recommender have potential to be disruptive in many parts of the music industry (not just end-user consumption)

• Creating truly personalized music RecSys and evaluating user satisfaction is still challenging
Recommended Reading

Spotify Teardown: 
Inside the Black Box of Streaming Music,

Maria Eriksson, Rasmus Fleischer, 
Anna Johansson, Pelle Snickars, and 
Patrick Vonderau.

Practical Resources: Toolboxes and Datasets
Toolboxes for RecSys (CF)

- MyMediaLite (C#): http://www.mymedialite.net
- scikit-surprise (Python): http://surpriselib.com
- Apache Mahout Recommenders (with Spark): http://mahout.apache.org
- Rival (Evaluation, Reproducibility; Java): http://rival.recommenders.net
- + any machine learning/linear algebra package
Practical: Toolboxes for Music Content Analysis

- **Essentia** (C++, Python): [http://essentia.upf.edu](http://essentia.upf.edu)
- **Librosa** (Python): [https://github.com/librosa](https://github.com/librosa)
- **madmom** (Python): [https://github.com/CPJKU/madmom](https://github.com/CPJKU/madmom)
- **Marsyas** (C++): [http://marsyas.info](http://marsyas.info)
- **jMIR** (Java): [http://jmir.sourceforge.net](http://jmir.sourceforge.net)
- **Sonic Visualiser** (MIR through VAMP plugins): [http://sonicvisualiser.org](http://sonicvisualiser.org)
Toolboxes for Text Analysis

- Natural Language Toolkit nltk (Python): https://www.nltk.org
- Gensim (Python): https://radimrehurek.com/gensim/
- GATE (Java): https://gate.ac.uk
- MeTA (C++): https://meta-toolkit.org
- Apache OpenNLP (Java): http://opennlp.apache.org
- jMIR (Java): http://jmir.sourceforge.net
Practical: Datasets

• Million Song Dataset: https://labrosa.ee.columbia.edu/millionsong
• Million Musical Tweets Dataset: http://www.cp.jku.at/datasets/mmttd
• #nowplaying Spotify playlists dataset: http://dbis-nowplaying.uibk.ac.at
• LFM-1b: http://www.cp.jku.at/datasets/LFM-1b
• Celma’s Last.fm datasets: http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/index.html
• Art of the Mix (AotM-2011) playlists: https://bmcfee.github.io/data/aotm2011.html