Finding a path through the Juke Box
The Playlist Tutorial

Ben Fields, Paul Lamere
ISMIR 2010
“I still maintain that music is the best way of getting the self-expression job done.”

Nick Hornby
Speakers

Ben Fields
Goldsmiths University of London
benfields.net

Paul Lamere
The Echo Nest
MusicMachinery.com
Goals

• Understand where and why playlists are important

• Understand current and past methods of playlist construction

• Understand the whys and hows of various evaluation methods
Introduction
What is a playlist?

- mixtape
- prerecorded DJ set/mix CD
- live DJ set (typically *mixed*)
- radioshow logs
- an album
- functional music (eg. Muzak)
- any ordered list of songs?
What is a playlist?

We define a playlist as a set of songs meant to be listened to as a group, usually with an explicit order.
Why is playlisting important?

- Ultimately, music is consumed through listening
- An awareness of this act of listening is critical to successful MIR application
- The playlist is a formalization of this listening process
- Playlists have a traditional revenue model for artists and labels (e.g. radio)
Brief History of Playlists
Mixed Concert Programs

• Marks the beginnings international combinations of music from multiple composers

• Begins circa 1850 in London

• The idea of a set of music being curated begins to form
Early Broadcast Media

- Moving the ethos of the earlier period onto the radio
- Biggest changes are technology
  - Broadcast = larger simultaneous audience
  - Phonograph brings recorded music
- Initial broadcasts (eg. 1906 - Fessenden) as publicity stunts
- First continuous broadcast 1920 - Frank Conrad
Rock On the Radio

• Radio as a medium begins to push certain genres, especially rock and roll and r ‘n’ b

• Playlist first used to describe (unordered) sets of songs

• Personality driven
  • John Peel
  • Casey Kasem
Disco & Hip-Hop
emergence of the club DJ

- DJs at disco nightclubs, with a mixer and two turntables, saw the birth of the idea of **continuous mixing**

- DJs wanted dancers to not notice song transitions, and techniques such as **beat matching** and **phrase alignment** were pioneered

- Hip-Hop saw this idea pushed further, as DJs became live remixers, turning the turntable into an instrument

- At the same time, club DJs started to become the top billing over live acts, the curator becoming more of a draw than the artist
The Playlist Goes Personal

- The emergence of portable audio devices drives the popularity of cassette tapes
- This in turn leads to reordering and combining of disparate material into mixtapes
- Mixtapes themselves are traded and distributed socially, providing a means for recommendation and discovery
- In hip-hop, mixtapes served as the first recordings of new DJs featuring novel mixes and leading to current phenomenon of Mix [CD|set|tape] (now on CD or other digital media)
Now With Internet

• The Web’s increase in popularity and MP3 audio compression allow for practical sharing of music of the Internet
• This brings the mixtape for physical sharing to non-place sharing.
• Streaming-over-internet radio emerges
• Playlists on the cloud: play.me, spotify, etc.
Aspects of a good playlist
Aspects of a good Playlist

To me, making a tape is like writing a letter — there's a lot of erasing and rethinking and starting again. A good compilation tape, like breaking up, is hard to do. You've got to kick off with a corker, to hold the attention (...), and then you've got to up it a notch, or cool it a notch, and you can't have white music and black music together, unless the white music sounds like black music, and you can't have two tracks by the same artist side by side, unless you've done the whole thing in pairs and...oh, there are loads of rules. - Nick Hornby, High Fidelity
Factors affecting a good playlist

- The **songs** in the playlist
  - Listener’s **preference** for the songs
  - Listener’s **familiarity** with the songs
- Song **coherence**
- Artist / Song **variety**
- And more: **freshness, coolness**,
- The **order** of the songs:
  - The song **transitions**
  - Overall playlist **structure**
  - **Serendipity**
- The **context**
Factors affecting a good playlist

Figure 1: Importance of various factors in creating a playlist.

Survey with 14 participants
Factors affecting a good playlist

Figure 1: Importance of various factors in creating a playlist.

Survey with 14 participants
Factors affecting a good playlist

Figure 1: Importance of various factors in creating a playlist.

Survey with 14 participants
Factors affecting a good playlist

Figure 1: Importance of various factors in creating a playlist.

Survey with 14 participants
Factors affecting a good playlist

Figure 1: Importance of various factors in creating a playlist.

Survey with 14 participants
Factors affecting a good playlist

Figure 1: Importance of various factors in creating a playlist.

Survey with 14 participants
Factors affecting a good playlist

Figure 1: Importance of various factors in creating a playlist.

Survey with 14 participants
Factors affecting a good playlist

Songs in the playlist: 8
Transitions between songs: 7
Combination of genres: 6
Combination of artists: 5
Structure (e.g. song order): 4
Variation/coherence: 3
First song: 2
Last song: 1

Figure 1: Importance of various factors in creating a playlist.

Survey with 14 participants
Factors affecting preference

- **Musical taste** - long term slowly evolving commitment to a genre
- Recent listening **history**
- **Mood** or state of mind
- The **context**: listening, driving, studying, working, exercising, etc.
- The **familiarity**
  - People sometimes prefer to listen to the familiar songs that they like less than non-familiar songs
  - Familiarity significantly predicts choice when controlling for the effects of liking, regret, and ‘coolness’
Coherence

Organizing principals for mix help requests

• Artist / Genre / Style
• Song similarity
• Event or activity
• Romance
• Message or story
• Mood
• Challenge or puzzle
• Orchestration
• Characteristic of the mix recipient
• Cultural references
Coherence

Organizing principals for mix help requests

- Artist / Genre / Style
- Song similarity
- Event or activity
- Romance
- Message or story
- Mood
- Challenge or puzzle
- Orchestration
- Characteristic of the mix recipient
- Cultural references

“acoustic-country-folk type stuff”
Coherence
Organizing principals for mix help requests

- Artist / Genre / Style
  “acoustic-country-folk type stuff”,
- Song similarity
- Event or activity
  “anti-Valentine mix”
- Romance
- Message or story
- Mood
- Challenge or puzzle
- Orchestration
- Characteristic of the mix recipient
- Cultural references
Coherence

Organizing principals for mix help requests

- Artist / Genre / Style
  - "acoustic-country-folk type stuff"
- Song similarity
  - "anti-Valentine mix"
- Event or activity
- Romance
- Message or story
- Mood
- Challenge or puzzle
- Orchestration
- Characteristic of the mix recipient
- Cultural references

"More of an Art than a Science": Supporting the Creation of Playlists and Mixes
Sally Jo Cunningham, David Bainbridge, Annette Falconer
Coherence

Organizing principals for mix help requests

- Artist / Genre / Style
  
- Song similarity

- Event or activity

- Romance

- Message or story

- Mood

- Challenge or puzzle

- Orchestration

- Characteristic of the mix recipient

- Cultural references

“acoustic-country-folk type stuff”,

“anti-Valentine mix”

a mix with the title “‘quit being a douche’, ’cause I’m in love with you.”

song whose title is a question?

‘More of an Art than a Science’: Supporting the Creation of Playlists and Mixes
Sally Jo Cunningham, David Bainbridge, Annette Falconer
Coherence
Organizing principals for mix help requests

- Artist / Genre / Style
  "acoustic-country-folk type stuff",
- Song similarity
- Event or activity
  "anti-Valentine mix"
- Romance
- Message or story
  a mix with the title "‘quit being a douche’, ’cause I’m in love with you.”
- Mood
- Challenge or puzzle
- Orchestration
- Characteristic of the mix recipient
- Cultural references

"song whose title is a question?"
"songs where the singer hums for a little bit"
Coherence

Organizing principals for mix help requests

- Artist / Genre / Style
  - “acoustic-country-folk type stuff”
- Song similarity
  - “anti-Valentine mix”
- Event or activity
- Romance
  - a mix with the title “‘quit being a douche’, ’cause I’m in love with you.”
- Message or story
- Mood
  - song whose title is a question?
- Challenge or puzzle
  - songs where the singer hums for a little bit
- Orchestration
  - “songs about superheroes”
- Characteristic of the mix recipient
- Cultural references

‘More of an Art than a Science’: Supporting the Creation of Playlists and Mixes
Sally Jo Cunningham, David Bainbridge, Annette Falconer
“People have gotten used to listening to songs in the order they want, and they'll want to continue to do so even if they can't get the individual songs from file-trading programs.”

Phil Leigh
Ordering Principals

- Bucket of similars, genre
- Acoustic attributes such as tempo, loudness, danceability
- Social attributes such as popularity, ‘hotness’
- Mood attributes (‘sad’ to ‘happy’)
- Theme / lyrics
- Alphabetical
- Chronological
- Random
- Song transitions
- Novelty orderings
## Novelty ordering

<table>
<thead>
<tr>
<th></th>
<th>Song Title</th>
<th>Artist</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>We Wish You A Merry Christmas</td>
<td>Weezer</td>
</tr>
<tr>
<td>1</td>
<td>Stranger Things Have Happened</td>
<td>Foo Fighters</td>
</tr>
<tr>
<td>2</td>
<td>Dude We're Finally Landing</td>
<td>Rivers Cuomo</td>
</tr>
<tr>
<td>3</td>
<td>Gotta Be Somebody's Blues</td>
<td>Jimmy Eat World</td>
</tr>
<tr>
<td>4</td>
<td>Someday You Will Be Loved</td>
<td>Death Cab For Cutie</td>
</tr>
<tr>
<td>5</td>
<td>Dancing In The Moonlight</td>
<td>The Smashing Pumpkins</td>
</tr>
<tr>
<td>6</td>
<td>Take The Long Way Round</td>
<td>Teenage Fanclub</td>
</tr>
<tr>
<td>7</td>
<td>Don't Make Me Prove It</td>
<td>Veruca Salt</td>
</tr>
<tr>
<td>8</td>
<td>The Sacred And Profane</td>
<td>Smashing Pumpkins, The</td>
</tr>
<tr>
<td>9</td>
<td>Everything Is Alright</td>
<td>Motion City Soundtrack</td>
</tr>
<tr>
<td>10</td>
<td>Trains, brains &amp; rain</td>
<td>The Flaming Lips</td>
</tr>
<tr>
<td>11</td>
<td>No One Needs To Know</td>
<td>Ozma</td>
</tr>
<tr>
<td>12</td>
<td>What Is Your Secret</td>
<td>Nada Surf</td>
</tr>
<tr>
<td>13</td>
<td>The Spark That Bled</td>
<td>Flaming Lips, The</td>
</tr>
<tr>
<td>14</td>
<td>Defending The Faith</td>
<td>Nerf Herder</td>
</tr>
</tbody>
</table>
Novelty ordering

0  We Wish You A Merry Christmas - Weezer
1  Stranger Things Have Happened - Foo Fighters
2  Dude We're Finally Landing - Rivers Cuomo
3  Gotta Be Somebody's Blues - Jimmy Eat World
4  Someday You Will Be Loved - Death Cab For Cutie
5  Dancing In The Moonlight - The Smashing Pumpkins
6  Take The Long Way Round - Teenage Fanclub
7  Don't Make Me Prove It - Veruca Salt
8  The Sacred And Profane - Smashing Pumpkins, The
9  Everything Is Alright - Motion City Soundtrack
10 Trains, brains & rain - The Flaming Lips
11 No One Needs To Know - Ozma
12 What Is Your Secret - Nada Surf
13 The Spark That Bled - Flaming Lips, The
14 Defending The Faith - Nerf Herder
Nothing Ordering

0  We Wish You A Merry Christmas - Weezer
1  Stranger Things Have Happened - Foo Fighters
2  Dude We're Finally Landing - Rivers Cuomo
3  Gotta Be Somebody's Blues - Jimmy Eat World
4  Someday You Will Be Loved - Death Cab For Cutie
5  Dancing In The Moonlight - The Smashing Pumpkins
6  Take The Long Way Round - Teenage Fanclub
7  Don't Make Me Prove It - Veruca Salt
8  The Sacred And Profane - Smashing Pumpkins, The
9  Everything Is Alright - Motion City Soundtrack
10 Trains, brains & rain - The Flaming Lips
11 No One Needs To Know - Ozma
12 What Is Your Secret - Nada Surf
13 The Spark That Bled - Flaming Lips, The
14 Defending The Faith - Nerf Herder
Novelty ordering

0  We Wish You A Merry Christmas - Weezer
1  Stranger Things Have Happened - Foo Fighters
2  Dude We're Finally Landing - Rivers Cuomo
3  Gotta Be Somebody's Blues - Jimmy Eat World
4  Someday You Will Be Loved - Death Cab For Cutie
5  Dancing In The Moonlight - The Smashing Pumpkins
6  Take The Long Way Round - Teenage Fanclub
7  Don't Make Me Prove It - Veruca Salt
8  The Sacred And Profane - Smashing Pumpkins, The
9  Everything Is Alright - Motion City Soundtrack
10 Trains, brains & rain - The Flaming Lips
11 No One Needs To Know - Ozma
12 What Is Your Secret - Nada Surf
13 The Spark That Bled - Flaming Lips, The
14 Defending The Faith - Nerf Herder
Novelty ordering

0  We Wish You A Merry Christmas - Weezer
1  Stranger Things Have Happened - Foo Fighters
2  Dude We're Finally Landing - Rivers Cuomo
3  Gotta Be Somebody's Blues - Jimmy Eat World
4  Someday You Will Be Loved - Death Cab For Cutie
5  Dancing In The Moonlight - The Smashing Pumpkins
6  Take The Long Way Round - Teenage Fanclub
7  Don't Make Me Prove It - Veruca Salt
8  The Sacred And Profane - Smashing Pumpkins, The
9  Everything Is Alright - Motion City Soundtrack
10 Trains, brains & rain - The Flaming Lips
11 No One Needs To Know - Ozma
12 What Is Your Secret - Nada Surf
13 The Spark That Bled - Flaming Lips, The
14 Defending The Faith - Nerf Herder
Where song order rules
The Dance DJ

• For the Dance DJ - song order and transitions are especially important

• Primary goal: make people dance

• How?
  • Selecting
    • tracks that mix well
    • takes the audience on a journey
    • audience feedback is important
  • Mixing
    • seamless song transitions

Hang the DJ: Automatic Sequencing and Seamless Mixing of Dance-Music Tracks
Dave Cliff Publishing Systems and Systems Laboratory HP Laboratories Bristol HPL-2000-104 9th August, 2000*

By BRENT SILBY
Tempo Trajectories

Warmup

Cool down

Nightclub
Coherence
Song to Song

Beat Matching and Cross-fading

hpDJ: An automated DJ with floorshow feedback
Dave Cliff Digital Media Systems Laboratory HP Laboratories Bristol
Don’t underestimate the power of the shuffle
Don’t underestimate the power of the shuffle

*laugh-out-loud pleasurable*
Don’t underestimate the power of the shuffle

white-knuckle ride
Don’t underestimate the power of the shuffle

“...teaches me connections between disparate kinds of music and the infinite void. I understand the universe better”
Don’t underestimate the power of the shuffle

...forge(ing) new connections
between my heart and my ears
Don’t underestimate the power of the shuffle

each randomly-sequenced track like an aural postcard
Don’t underestimate the power of the shuffle

had made me re-examine things I thought I knew about my favourite music
Don’t underestimate the power of the shuffle

...hear(ing) songs that I haven’t heard for years and fall (ing) in love with them again
Don’t underestimate the power of the shuffle

Random shuffle can turn large music libraries into an ‘Aladdin’s cave’ of aural surprises
Don’t underestimate the power of the shuffle

...the random effect delivers a sequence of music so perfectly thematically 'in tune' that (it) is quite unsettling
Serendipity of the shuffle

Finding meaningful experience in chance encounters

- Serendipity can improve the listening experience
- Choosing songs randomly from a personal collection can yield serendipitous listening
- Drawing from too large, or too small of a collection reduces serendipity
People like shuffle play

<table>
<thead>
<tr>
<th>Preferred listening</th>
<th>content organisation</th>
<th>unconstrained</th>
<th>constrained</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>shuffle</td>
<td>22</td>
<td>69</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>both</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>sequential</td>
<td>13</td>
<td>1</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>74</td>
<td>113</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Preferred listening mode (shuffle or sequential) and organisation of music content (constrained or unconstrained)

People shuffle genres, albums and playlists
Playlist tradeoffs

Variety ↔ Coherence

Freshness ↔ Familiarity

Surprise ↔ Order

Different listeners have different optimal settings. Mood and context can affect optimal settings.
## Playlist Variety

A good playlist is not a bag of similar tracks

<table>
<thead>
<tr>
<th>#</th>
<th>Track</th>
<th>Album</th>
<th>Artist</th>
<th>Genre</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Farrakorn</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>2</td>
<td>What's Wrong with my foot?</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>3</td>
<td>I love her to Pieces</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>4</td>
<td>In my livid eyes</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>5</td>
<td>A little exposure</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>6</td>
<td>Donkey Punch</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>7</td>
<td>Wow!</td>
<td>Gimme Some</td>
<td>Nova Express</td>
<td>Punk</td>
</tr>
<tr>
<td>8</td>
<td>Flowers on the Wall</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>9</td>
<td>Wet Brain</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>10</td>
<td>Tammy ate a bad piece of pork</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>11</td>
<td>Pucker String</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>12</td>
<td>Pizzle: Party Patrol</td>
<td>High Energy Rock and Roll</td>
<td>Magnatune Compilation</td>
<td>Rock</td>
</tr>
<tr>
<td>13</td>
<td>Nunchukkaboot</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>14</td>
<td>Party Patrol</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>15</td>
<td>Motorway</td>
<td>Gimme Some</td>
<td>Nova Express</td>
<td>Punk</td>
</tr>
</tbody>
</table>
## Playlist Variety

A good playlist is not a bag of similar tracks

<table>
<thead>
<tr>
<th>#</th>
<th>Track</th>
<th>Album</th>
<th>Artist</th>
<th>Genre</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pizzle: In my livid eyes</td>
<td>High Energy Rock and Roll</td>
<td>Magnatune Compilation</td>
<td>Rock</td>
</tr>
<tr>
<td>2</td>
<td>In my livid eyes</td>
<td>Party Patrol</td>
<td>Pizzle</td>
<td>Punk</td>
</tr>
<tr>
<td>3</td>
<td>Wow!</td>
<td>Gimme Some</td>
<td>Nova Express</td>
<td>Punk</td>
</tr>
<tr>
<td>4</td>
<td>Euthanize Tunnel Zone</td>
<td>Hellavator Musick</td>
<td>Skitzo</td>
<td>Metal</td>
</tr>
<tr>
<td>5</td>
<td>Hostage Situation</td>
<td>Listen Up, Baby!</td>
<td>Electric Frankenstein</td>
<td>Punk</td>
</tr>
<tr>
<td>6</td>
<td>Dirty brown duster</td>
<td>Jacksploitation</td>
<td>Jackalopes</td>
<td>Punk</td>
</tr>
<tr>
<td>7</td>
<td>Park that ass</td>
<td>Geeking Dream</td>
<td>The Strap Ons</td>
<td>Punk</td>
</tr>
<tr>
<td>8</td>
<td>Higher education</td>
<td>Thrill Hype</td>
<td>The Napoleon Blown Apart</td>
<td>Punk Rock</td>
</tr>
<tr>
<td>9</td>
<td>KC rip off</td>
<td>Up from the mud</td>
<td>Spinecar</td>
<td>Hard Rock</td>
</tr>
<tr>
<td>10</td>
<td>As it Descends</td>
<td>Night of the Black Wyvern</td>
<td>Utopia Banished</td>
<td>Metal</td>
</tr>
<tr>
<td>11</td>
<td>No Cure</td>
<td>8 Seconds</td>
<td>Pain Factor</td>
<td>Metal</td>
</tr>
<tr>
<td>12</td>
<td>Everyday Like Saturday (bonu...</td>
<td>Middle Age Suicide</td>
<td>Rocket City Riot</td>
<td>Rock</td>
</tr>
<tr>
<td>13</td>
<td>Function</td>
<td>Trancelucent</td>
<td>Somadrome</td>
<td>Rock</td>
</tr>
<tr>
<td>14</td>
<td>Feverdream #1</td>
<td>Alpha &amp; Oranges</td>
<td>Atomic Opera</td>
<td>Hard Rock</td>
</tr>
<tr>
<td>15</td>
<td>Look And Feel Years Younger</td>
<td>I Don't Know What I'm Doing</td>
<td>Brad Sucks</td>
<td>Rock</td>
</tr>
</tbody>
</table>
Playlist Variety
A good playlist is not a bag of similar tracks
Playlisting is not Recommendation
Playlisting is not Recommendation

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primarily for music discovery</td>
<td>Primarily for music listening</td>
</tr>
<tr>
<td>Minimize familiar artists</td>
<td>Familiar artists in abundance</td>
</tr>
<tr>
<td>Order not important</td>
<td>Order can be critical</td>
</tr>
<tr>
<td>Limited Context (shopping)</td>
<td>Rich contexts - party, jogging, working, gifts</td>
</tr>
</tbody>
</table>
### Playlisting is not Recommendation

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primarily for music discovery</td>
<td>Primarily for music listening</td>
</tr>
</tbody>
</table>
# Playlisting is not Recommendation

<table>
<thead>
<tr>
<th><strong>Recommendation</strong></th>
<th><strong>Playlist</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Primarily for music discovery</td>
<td>Primarily for music listening</td>
</tr>
<tr>
<td>Minimize familiar artists</td>
<td>Familiar artists in abundance</td>
</tr>
</tbody>
</table>
Playlisting is not Recommendation

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primarily for music discovery</td>
<td>Primarily for music listening</td>
</tr>
<tr>
<td>Minimize familiar artists</td>
<td>Famililiar artists in abundance</td>
</tr>
<tr>
<td>Order not important</td>
<td>Order can be critical</td>
</tr>
</tbody>
</table>
# Playlisting is not Recommendation

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primarily for music discovery</td>
<td>Primarily for music listening</td>
</tr>
<tr>
<td>Minimize familiar artists</td>
<td>Familiar artists in abundance</td>
</tr>
<tr>
<td>Order not important</td>
<td>Order can be critical</td>
</tr>
<tr>
<td>Limited Context (shopping)</td>
<td>Rich contexts - party, jogging, working, gifts</td>
</tr>
</tbody>
</table>
Playlisting is not Recommendation

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primarily for music discovery</td>
<td>Primarily for music listening</td>
</tr>
<tr>
<td>Minimize familiar artists</td>
<td>Familiar artists in abundance</td>
</tr>
<tr>
<td>Order not important</td>
<td>Order can be critical</td>
</tr>
<tr>
<td>Limited Context (shopping)</td>
<td>Rich contexts - party, jogging, working, gifts</td>
</tr>
</tbody>
</table>

However, playlists may be better vector for music discovery than traditional recommendation
Playlisting nuts and bolts formats and rules
Playlist formats

• Lots of formats - Some notable examples:
  • M3U - simple list of files - one per line
  • XSPF - ‘spiff’ - XML based format
  • The Playback Ontology

• Resources:
  • http://microformats.org/wiki/audio-info-formats
  • http://lizzy.sourceforge.net/docs/formats.html
  • http://gonze.com/playlists/playlist-format-survey.html
Example XSPF

```xml
<?xml version="1.0" encoding="UTF-8"?>
<playlist version="1" xmlns="http://xspf.org/ns/0/">
  <trackList>
    <track>
      <location>http://example.com/song_1.mp3</location>
      <creator>Led Zeppelin</creator>
      <album>Houses of the Holy</album>
      <title>No Quarter</title>
      <annotation>I love this song</annotation>
      <duration>271066</duration>
      <image>http://images.amazon.com/images/P/B000002J0B.jpg</image>
      <info>http://example.com</info>
    </track>
    <track>
      <location>http://example.com/song_1.mp3</location>
      <creator>Led Zeppelin</creator>
      <album>ii</album>
      <title>No Quarter</title>
      <annotation>This one too</annotation>
      <duration>271066</duration>
      <image>http://images.amazon.com/images/P/B000002J0B.jpg</image>
      <info>http://example.com</info>
    </track>
  </trackList>
</playlist>
```
The Playback Ontology

The Play Back Ontology provides basic concepts and properties for describing concepts that are related to the play back domain, e.g. a playlist, play back and skip counter, on/ for the Semantic Web.
The Playback Ontology

Modeling items in the playlist by extending the ordered list ontology

The Playback Ontology

Expressing similarity and creation provenance

http://smiy.sourceforge.net/pbo/spec/playbackontology.html
Survey of playlisting systems and tools
Manual Non-Social
Rush: Repeated Recommendations on Mobile Devices

Dominikus Baur, Sebastian Boring, Andreas Butz
Playlist creation tools
Playlist creation tools
Do people use Smart Playlists?
Do people use Smart Playlists?

62% of polled have 5 or less smart playlists

Informal poll with 162 respondents
Automated Non-Social iTunes Genius Mix

- Alternative Pop/Rock Mix
- Brit-Pop & Rock Mix
- Adult Alternative Mix
- Classic Rock Mix
- Punk Mix
- Chamber Pop Mix
- Jazz Mix
- Indie Rock & Lo-Fi Mix
- Singer/Songwriter Mix
- Traditional Folk Mix
- Electro-Pop Mix
- Alt Metal Mix
Automated Non-Social
Automated Non-Social
Mood Agent

- Use sliders to set levels of 5 ‘moods’:
  - Sensual
  - Tender
  - Happy
  - Angry
  - Tempo
AMG tapestry
Visual Playlist Generation on the Artist Map

Van Gulick, Vignoli
Path From Kanye West To Taylor Swift

1. Say You Will
   *Kanye West*

2. It's A New Day
   *will.i.am*

3. Fergalicious
   *Fergie*

4. You Are What You Are (Beautiful)
   *Christina Aguilera*

5. Love Story
   *Taylor Swift*
GeoMuzik: A geographic interface for large music collections: Òscar Celma, Marcelo Nunes
Using visualizations to build playlists

MusicBox: Mapping and visualizing music collections
Anita Lillie’s Masters Thesis at the MIT Media Lab
Search Inside the Music

Using 3D visualizations to explore and discover music.
Paul Lamere and Doug Eck
Automated Social
Last.fm

Your Friends
Pending Friend Requests (11)
gearmonkey

Friends listening now
ocelma
hiqlokey
Âme – Junggesellenmaschine  yesterday evening
Automated Social

Radio Paradise

92 comments for this song: Log in above to post your comment

nagsheadlocal
(North Carolina, the new New Jersey)
Posted: Jun 24, 2010 - 05:28
Man, do I love a drummer who can mix times and alternate between the upbeat and the downbeat. I'm sitting here nodding at work.

shutter
(You can't get here from there)
Posted: Jun 02, 2010 - 10:55
yeah, way cool tune.

lysisphere
(largest contiguous ponderosa forest)
Posted: Jun 02, 2010 - 10:51
That was a sweet transition Bill!
DMCA Radio

US rules for Internet streaming radio

• In a single 3 hour period:
  • No more than three songs from the same recording
  • No more than two songs in a row, from the same recording
  • No more than four songs from the same artist or anthology
  • No more than three songs in a row from the same artist or anthology

Note that there are no explicit rules that limit skipping
Terrestrial Radio Programming
### Terrestrial Radio Programming

#### On Air Playlist

<table>
<thead>
<tr>
<th>Song</th>
<th>Artist</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lake of Fire</td>
<td>Nirvana</td>
<td>8:39am</td>
</tr>
<tr>
<td>T.N.T.</td>
<td>AC/DC</td>
<td>8:36am</td>
</tr>
<tr>
<td>People of the Sun</td>
<td>Rage Against the Machine</td>
<td>8:33am</td>
</tr>
<tr>
<td>In One Ear</td>
<td>Cage the Elephant</td>
<td>8:30am</td>
</tr>
<tr>
<td>Bad</td>
<td>U2</td>
<td>8:24am</td>
</tr>
<tr>
<td>Black</td>
<td>Pearl Jam</td>
<td>8:18am</td>
</tr>
<tr>
<td>Bad Company</td>
<td>Five Finger Death Punch</td>
<td>8:08am</td>
</tr>
<tr>
<td>Livin' on the Edge</td>
<td>Aerosmith</td>
<td>8:02am</td>
</tr>
</tbody>
</table>
Radio station programming rules

• Divide the day into a set of 5 (typically) ‘dayparts’: Mid-6A, 6A-10A, 10A-3P, 3P-7P, and 7P-12Mid

• For each daypart:
  • Gender, Tempo, Intensity, Mood, Style controls
  • Artist separation controls [global and individual artist]
  • Prior-day horizontal title separation
  • Artist blocks [multiple songs in-a-row by same artist]
  • "Never-Violate" and "Preferred" rules
  • Hour circulation rules
Automated Radio Programming

[Image of software interface for setting minimum separation rules for artists]
Automated Radio Programming

![Sample Edit Rule for RuleSet 1]

- **Rule Type**: Gender
- **Restriction Type**: Max-in-a-Row
- **Restricted Code**: F-Female

**Gender F-Female Max-in-a-Row**

- **Preferred Rule**: # Songs
- **Preferred Violation Points**: 0 to 1000 Points
- **Never-Violate Rule**: # Songs

The maximum # of songs with Gender code F-Female which may appear without a different Gender code appearing. Enter # songs in either or both of the PREFERRED and/or NEVER-VIOLATE boxes above.

---

**Natural Broadcast Systems**

Reliable, High Quality Broadcast Management Software

At Prices That Make Sense!
Automated Radio Programming
# Automated Radio Programming

The image displays a software interface for automated radio programming. The interface is titled "Sample DayPart Codes" and includes a table with columns for DayPart ID, DayPart Name, Action on Error, and Song Count. The table entries include:

- **DayPart ID**: A, B, C, D, E
- **DayPart Name**: No Drives Or Prime, No Daytime At All, Saturday Only Cruise, Cruising Only, No Weekday Middays
- **Action on Error**: Rotate Song
- **Song Count**: 167, 41, 18, 42, 2

Additionally, there is a grid layout for scheduling song placements with symbols indicating whether a song is allowed or prohibited. The symbols include an 'X' for prohibited and a 'L' for allowed, with red and black colors.

At the bottom of the image, there is an advertisement for Natural Broadcast Systems, which provides reliable, high-quality broadcast management software at prices that make sense.
Automated Social PartyStrands
art of the mix

- Hand made playlists
- Mix art
- Web services
- Pre-crawled data at:

- Browse / search for playlists
- Create a playlist:
  - Search for artist / songs
  - Add songs to a playlist
  - Re-order the playlist
  - Describe the playlist:
    - title, description, tags
  - Decorate the playlist
  - Publish the playlist
The place to discover and listen to free music online, create free playlists, and share it all on your favorite social networks: Facebook, Twitter, Blogger, and more.

Start building a playlist

Search for a song or artist

Today's Top Searches
- eminem
- lil wayne
- drake
- 3oh!3
- justin bieber
- usher
- lady gaga
- b.o.b
- katy perry
- taylor swift
- kesha
- airplanes
- t.i.
- nicki minaj
- nickelback
- trey songz
- linkin park
- glee
- beyonce
- miley cyrus
Create A Music Playlist!
Your music says a lot about you.

Enter an artist, a song title or both...  

Search

Popular Searches:
Mike Posner, Katy Perry, Taio Cruz, BoB, Jason Derülo, Eminem, Drake, Eminem, Usher, Travie McCoy

Share your playlist on MySpace, Facebook, Friendster, hi5, Bebo, blogs & more.

Sign Up!

Social Playlists

**Funk Music** (25 tracks)
1. Play That Funky Music – W ...
2. Stevie Wonder "SUPER ... 
3. Super Freak By Rick James
4. Brick – Dazz
5. Give Up The Funk By Parli ... and 20 more

Popular Songs

- Justin Bieber – Baby Ft. Ludacris
  - 31,068 listeners
- Sean Kingston & Justin Bieber "eerie Meenie"
  - 17,540 listeners
- Not Afraid By Eminem
  - 10,658 listeners
Spotify

- Sharable playlists
- Collaborative playlists
- Many 3rd party playlist sites
Spotify

- Sharable playlists
- Collaborative playlists
- Many 3rd party playlist sites
Spotify

- Sharable playlists
- Collaborative playlists
- Many 3rd party playlist sites
Spotify

- Sharable playlists
- Collaborative playlists
- Many 3rd party playlist sites
Spotify

- Sharable playlists
- Collaborative playlists
- Many 3rd party playlist sites
Spotify

- Sharable playlists
- Collaborative playlists
- Many 3rd party playlist sites

There are currently 23,659 Spotify Playlists ready for you to discover!
Mix Enablers
mixcloud
Mix Enablers
mixcloud

- Free social networking platform organized around the exchange of long form audio, principally [dance] music
- Provides a means for DJs (aspiring and professional) to connect with the audience and into the Web of Things
Mix Enablers

mixlr
Mix Enablers
mixlr

- Focused on adding social features to centralized multicasting
- Supports live and recorded (mixed and unmixed) streams
- Social connectivity is web-based, broadcaster is a native application
- Native app provides integration with common DJ tools

Artist
Emerson, Lake & Palmer  Artist statistics  Add setlist
Venue
Victoria Park, London, England
Attendees
aeolist cafcchegs Steban14
Last edited
July 31, 2010 2:04:17 PM UTC by Blackadder

1. Karn Evil 9: 1st Impression, Part 2
2. The Barbarian
3. Bitches Crystal
4. Touch and Go
5. Knife-Edge
6. From The Beginning
7. Take a Pebble
8. Tarkus
9. Farewell to Arms
10. Lucky Man
11. Pictures at an Exhibition
12. Fanfare for the Common Man/Drum Solo/Rondo
setlist.fm

A wiki for concert setlists

They have an API!


Artist
Emerson, Lake & Palmer

Venue
Victoria Park, London, England

Attendees
aeolist cabccheqs Steban14

Last edited
July 31, 2010 2:04:17 PM UTC by Blackadder

1. Karn Evil 9: 1st Impression, Part 2
2. The Barbarian
3. Bitches Crystal
4. Touch and Go
5. Knife-Edge
6. From The Beginning
7. Take a Pebble
8. Tarkus
9. Farewell to Arms
10. Lucky Man
11. Pictures at an Exhibition
12. Fanfare for the Common Man/Drum Solo/Rondo

REST Endpoints

/0.1/artist/{mbid}
/0.1/city/{geoid}
/0.1/search/artists
/0.1/search/cities
/0.1/search/countries
/0.1/search/setlists
/0.1/search/venues
/0.1/setlist/{setlistId}
/0.1/venue/{venueId}
/0.1/artist/{mbid}/setlists
/0.1/setlist/lastFm/{lastFmEventId}
/0.1/setlist/version/{versionId}
/0.1/venue/{venueId}/setlists
/0.1/artist/{mbid}/tour/{tour}
The playlisting dead pool
research systems
Human-Facilitating Systems
Personal Radio

- An early collaborative filtering system
- Users rated songs directly
- Playlists are built by finding similar (via Pearson’s correlation coefficient) users
- Playlists can, once built, be streamed, named, shared and modified
- Order is either random or user defined

User ratings are gathered using explicit and implicit user feedback. Users can explicitly choose to rate individual track items or individual programmes on a scale of 1–5, where 5 is the top score. Implicit feedback is important for the less interactive or casual user. The reasoning behind this is that the casual user wants to interact as little as possible with the system and will intervene only when the system gets things badly wrong. In terms of implicit feedback the system allocates a score of 4 to recommended programmes and constituent items that have been saved to a user’s profile, or to programmes that have been built from scratch by the user. Programmes can be put together in a matter of seconds by simple clicking on the desired music item and adding it to the current playlist (See figure 1). Music items are indexed currently by genre, which is not satisfactory since the genre feature is not finely grained enough to capture the nuances within music types. Further research is planned on how to allow user indexing of music assets.

Once a programme has been built it can be played immediately and is automatically saved to the users profile for future retrieval. Programmes that are played more than three times are awarded the top score of 5, even though the average rating of constituent items may be lower. Our theory is that a well chosen collection of music has greater value than the sum of its constituent items. For one thing, there is some work involved in putting together a programme so there is some value in choosing something “off the shelf”. For another, a collection of music may contain the difficult to quantify feature of “mood” which depends on the collected items being played together. This feature is apparent where users amend their ratings for individual items as they appear in different programmes. Figure 2 illustrates an excerpt for the programme mellow and jazzy in which the user cchayes has rated four out of the five shown items. If cchayes chooses mellow and jazzy again he will be shown his ratings for the individual items within the programme and he may recast his vote. This facility is important because music taste does shift, and user profiles will have to move to reflect this. It is entirely probable that a user will cease to become a recommender in one neighbourhood only to have moved to another.
Personal Radio

- An early collaborative filtering system
- Users rated songs directly
- Playlists are built by finding similar (via Pearson’s correlation coefficient) users
- Playlists can, once built, be streamed, named, shared and modified
- Order is either random or user defined
Collaborative Choice

A public voting system
Collaborative Choice

Decentralized supply

Jukola: democratic music choice in a public space
Playlist Sharing

- Music should help convey status information and implicit presence
- Music should help build interpersonal relationships
- A good individual listening experience should be supported
- Support smooth continuous use
Playlist Sharing

1. Members associate music from their personal library to their activities and locations

2. For each new song, the system picks a random user and a song from that user’s current state

3. Music is streamed to each mobile device

4. The device displays the current song and which user assigned it
Field Tested:

- Music should help convey status information and implicit presence
- Music should help build interpersonal relationships
- A good individual listening experience should be supported
- Support smooth continuous use
Field Tested:

• Music should help convey status information and implicit presence

• Music should help build interpersonal relationships

• A good individual listening experience should be supported

• Support smooth continuous use

"I am a weather guy. Happy music for sunny days so to speak."
Field Tested:

• Music should help convey status information and implicit presence

• Music should help build interpersonal relationships

• A good individual listening experience should be supported

• Support smooth continuous use

"I am a weather guy. Happy music for sunny days so to speak."

“'I made her a CD because I can’t stand her music.”
Field Tested:

- Music should help convey status information and implicit presence
  
  "I am a weather guy. Happy music for sunny days so to speak."

- Music should help build interpersonal relationships
  
  “I made her a CD because I can’t stand her music.”

- A good individual listening experience should be supported
  
  Participants report on hearing between 30% - 50% “bad songs”.

- Support smooth continuous use
Field Tested:

- Music should help convey status information and implicit presence
  
  "I am a weather guy. Happy music for sunny days so to speak."

- Music should help build interpersonal relationships

  “I made her a CD because I can’t stand her music.”

- A good individual listening experience should be supported

  Participants report on hearing between 30% - 50% “bad songs”.

- Support smooth continuous use

  At such occasions, they may turn off the service and switch to their own music library.
Implications

• Smooth integration with individual music listening to encourage continuous use

• Allow flexibility and cues to support self-expression and enable touch points

• Support ongoing relationships

• Counterbalance experiences of bad songs and misinterpretations
Fully Automatic Systems
Nearest Neighbors
Nearest Neighbors
Nearest Neighbors
Pure Content

• Uses MFCCs and finds $N$ nearest neighbors

• Forms a graph with all songs weighted by distance

• Playlist is created by finding the shortest weighted path covering $N$ songs
Pure Content

<table>
<thead>
<tr>
<th>Relevance</th>
<th>Average nr. of relevant songs in playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size 5</td>
</tr>
<tr>
<td>Same Genre</td>
<td>3.46</td>
</tr>
<tr>
<td>Same Artist</td>
<td>1.34</td>
</tr>
<tr>
<td>Same Album</td>
<td>1.11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relevance</th>
<th>Scheme</th>
<th>Average nr. of relevant songs in playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Size 5</td>
</tr>
<tr>
<td>Same Genre</td>
<td>Trajectory,1</td>
<td>3.26</td>
</tr>
<tr>
<td>Same Artist</td>
<td></td>
<td>1.08</td>
</tr>
<tr>
<td>Same Album</td>
<td></td>
<td>0.89</td>
</tr>
<tr>
<td>Same Genre</td>
<td>Trajectory,2</td>
<td>3.33</td>
</tr>
<tr>
<td>Same Artist</td>
<td></td>
<td>1.23</td>
</tr>
<tr>
<td>Same Album</td>
<td></td>
<td>1.01</td>
</tr>
<tr>
<td>Same Genre</td>
<td>Feedback</td>
<td>3.40</td>
</tr>
<tr>
<td>Same Artist</td>
<td></td>
<td>1.27</td>
</tr>
<tr>
<td>Same Album</td>
<td></td>
<td>1.05</td>
</tr>
</tbody>
</table>

Content-Based Playlist Generation: Exploratory Experiments
Beth Logan
### Pure Content

<table>
<thead>
<tr>
<th>Relevance</th>
<th>Average nr. of relevant songs in playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size 5</td>
</tr>
<tr>
<td>Same Genre</td>
<td>3.46</td>
</tr>
<tr>
<td>Same Artist</td>
<td>1.34</td>
</tr>
<tr>
<td>Same Album</td>
<td>1.11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relevance</th>
<th>Scheme</th>
<th>Average nr. of relevant songs in playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Size 5</td>
</tr>
<tr>
<td>Same Genre</td>
<td>Trajectory,1</td>
<td>3.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.89</td>
</tr>
<tr>
<td>Same Artist</td>
<td>Trajectory,2</td>
<td>3.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.01</td>
</tr>
<tr>
<td>Same Album</td>
<td>Feedback</td>
<td>3.40</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.05</td>
</tr>
</tbody>
</table>
## Pure Content

<table>
<thead>
<tr>
<th>Relevance</th>
<th>Average nr. of relevant songs in playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size 5</td>
</tr>
<tr>
<td>Same Genre</td>
<td>3.46</td>
</tr>
<tr>
<td>Same Artist</td>
<td>1.34</td>
</tr>
<tr>
<td>Same Album</td>
<td>1.11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Relevance</th>
<th>Scheme</th>
<th>Average nr. of relevant songs in playlist</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Size 5</td>
</tr>
<tr>
<td>Same Genre</td>
<td>Trajectory,1</td>
<td>3.26</td>
</tr>
<tr>
<td>Same Artist</td>
<td></td>
<td>1.08</td>
</tr>
<tr>
<td>Same Album</td>
<td></td>
<td>0.89</td>
</tr>
<tr>
<td>Same Genre</td>
<td>Trajectory,2</td>
<td>3.33</td>
</tr>
<tr>
<td>Same Artist</td>
<td></td>
<td>1.23</td>
</tr>
<tr>
<td>Same Album</td>
<td></td>
<td>1.01</td>
</tr>
<tr>
<td>Same Genre</td>
<td>Feedback</td>
<td>3.40</td>
</tr>
<tr>
<td>Same Artist</td>
<td></td>
<td>1.27</td>
</tr>
<tr>
<td>Same Album</td>
<td></td>
<td>1.05</td>
</tr>
</tbody>
</table>
Learning a Gaussian Process Prior for Automatically Generating Music Playlists
John C. Platt and Christopher J.C. Burges and Steven Swenson and Christopher Weare and Alice Zheng

Metadata Models

<table>
<thead>
<tr>
<th>Metadata Field</th>
<th>Example Values</th>
<th>Number of Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genre</td>
<td>Jazz, Reggae, Hip-Hop</td>
<td>30</td>
</tr>
<tr>
<td>Subgenre</td>
<td>Heavy Metal, I’m So Sad and Spaced Out</td>
<td>572</td>
</tr>
<tr>
<td>Style</td>
<td>East Coast Rap, Gangsta Rap, West Coast Rap</td>
<td>890</td>
</tr>
<tr>
<td>Mood</td>
<td>Dreamy, Fun, Angry</td>
<td>21</td>
</tr>
<tr>
<td>Rhythm Type</td>
<td>Straight, Swing, Disco</td>
<td>10</td>
</tr>
<tr>
<td>Rhythm Description</td>
<td>Frenetic, Funky, Lazy</td>
<td>13</td>
</tr>
<tr>
<td>Vocal Code</td>
<td>Instrumental, Male, Female, Duet</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 1: Music metadata fields, with some example values
Metadata Models

<table>
<thead>
<tr>
<th>Metadata Field</th>
<th>Example Values</th>
<th>Number of Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genre</td>
<td>Jazz, Reggae, Hip-Hop</td>
<td>30</td>
</tr>
<tr>
<td>Subgenre</td>
<td>Heavy Metal, I’m So Sad and Spaced Out</td>
<td>572</td>
</tr>
<tr>
<td>Style</td>
<td>East Coast Rap, Gangsta Rap, West Coast Rap</td>
<td>890</td>
</tr>
<tr>
<td>Mood</td>
<td>Dreamy, Fun, Angry</td>
<td>21</td>
</tr>
<tr>
<td>Rhythm Type</td>
<td>Straight, Swing, Disco</td>
<td>10</td>
</tr>
<tr>
<td>Rhythm Description</td>
<td>Frenetic, Funky, Lazy</td>
<td>13</td>
</tr>
<tr>
<td>Vocal Code</td>
<td>Instrumental, Male, Female, Duet</td>
<td>6</td>
</tr>
</tbody>
</table>

- Use Gaussian Process Regression to create playlists based on seed tracks
### Metadata Models

<table>
<thead>
<tr>
<th>Metadata Field</th>
<th>Example Values</th>
<th>Number of Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genre</td>
<td>Jazz, Reggae, Hip-Hop</td>
<td>30</td>
</tr>
<tr>
<td>Subgenre</td>
<td>Heavy Metal, I’m So Sad and Spaced Out</td>
<td>572</td>
</tr>
<tr>
<td>Style</td>
<td>East Coast Rap, Gangsta Rap, West Coast Rap</td>
<td>890</td>
</tr>
<tr>
<td>Mood</td>
<td>Dreamy, Fun, Angry</td>
<td>21</td>
</tr>
<tr>
<td>Rhythm Type</td>
<td>Straight, Swing, Disco</td>
<td>10</td>
</tr>
<tr>
<td>Rhythm Description</td>
<td>Frenetic, Funky, Lazy</td>
<td>13</td>
</tr>
<tr>
<td>Vocal Code</td>
<td>Instrumental, Male, Female, Duet</td>
<td>6</td>
</tr>
</tbody>
</table>

- Use Gaussian Process Regression to create playlists based on seed tracks
- Using Kernel Meta-Training algorithm on albums to select the priors
Metadata Models

<table>
<thead>
<tr>
<th></th>
<th>Playlist 1</th>
<th>Playlist 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seed 1</td>
<td>Eagles, The Sad Cafe</td>
<td>Eagles, Life in the Fast Lane</td>
</tr>
<tr>
<td>2</td>
<td>Genesis, More Fool Me</td>
<td>Eagles, Victim of Love</td>
</tr>
<tr>
<td>3</td>
<td>Bee Gees, Rest Your Love On Me</td>
<td>Rolling Stones, Ruby Tuesday</td>
</tr>
<tr>
<td>4</td>
<td>Chicago, If You Leave Me Now</td>
<td>Led Zeppelin, Communication Breakdown</td>
</tr>
<tr>
<td>5</td>
<td>Eagles, After The Thrill Is Gone</td>
<td>Creedence Clearwater, Sweet Hitch-hiker</td>
</tr>
<tr>
<td></td>
<td>Cat Stevens, Wild World</td>
<td>Beatles, Revolution</td>
</tr>
</tbody>
</table>

- Use Gaussian Process Regression to create playlists based on seed tracks
- Using Kernel Meta-Training algorithm on albums to select the priors
- Playlists are formed based on the maximum log likelihood from the selected seed song
• Use Gaussian Process Regression to create playlists based on seed tracks

• Using Kernel Meta-Training algorithm on albums to select the priors

• Playlists are formed based on the maximum log likelihood from the selected seed song
Use Gaussian Process Regression to create playlists based on seed tracks

Using Kernel Meta-Training algorithm on albums to select the priors

Playlists are formed based on the maximum log likelihood from the selected seed song

<table>
<thead>
<tr>
<th>Playlist Method</th>
<th>Number of Seed Songs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>KMT + GPR</td>
<td>42.9</td>
</tr>
<tr>
<td>Hamming + GPR</td>
<td>32.7</td>
</tr>
<tr>
<td>Hamming + No GPR</td>
<td>32.7</td>
</tr>
<tr>
<td>Random Order</td>
<td>6.3</td>
</tr>
</tbody>
</table>
• Use Gaussian Process Regression to create playlists based on seed tracks

• Using Kernel Meta-Training algorithm on albums to select the priors

• Playlists are formed based on the maximum log likelihood from the selected seed song

<table>
<thead>
<tr>
<th>Playlist Method</th>
<th>Number of Seed Songs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>KMT + GPR</td>
<td>42.9</td>
</tr>
<tr>
<td>Hamming + GPR</td>
<td>32.7</td>
</tr>
<tr>
<td>Hamming + No GPR</td>
<td>32.7</td>
</tr>
<tr>
<td>Random Order</td>
<td>6.3</td>
</tr>
</tbody>
</table>
Traveling Sales Playlist?

In this paper, we present an extension to the "wheel". From our point of view, users are seldom capable of paying attention to the whole music collection. Therefore, we propose a novel approach that is based on audio only.

The problem of playlist generation is treated as a network flow problem in \cite{1}. Given a start track and an end track in a song collection, the algorithm finds a path (of segments) through the network satisfying user-defined constraints, the meta-data of each track is transformed into a similarity measure. In \cite{9}, we augment an interface to music collections with terms obtained from the web. The interface generates a playlist based on the similarity of the terms and the meta-data (cf. \cite{15}). This approach is based on Mel Frequency Cepstrum Coefficients (MFCCs) computed on short-time audio segments. As proposed in \cite{3}, we calculated 19 MFCCs. In \cite{12}, audio-based as well as web-based genre classification are used for the task of style detection on a set of 5 genres with 5 artists each. Combining the predictions made by both methods linearly yields perfect overall prediction for all test cases. In \cite{4}, audio-based track similarity is linearly combined with web-based artist similarity to obtain a new genre classification. From our point of view, users are seldom capable of paying attention to the whole music collection. Therefore, we propose a novel approach that is based on audio only.

Combining Audio-based Similarity with Web-based Data to Accelerate Automatic Music Playlist Generation
Peter Knees, Tim Pohle, Markus Schedl, and Gerhard Widmer
Traveling Sales Playlist?

• Using a combination of content-based song and web-based artist similarity to generate a distance matrix

• Approximation of TSP is used to find ‘tours’ through the collection

• Tested on two collections of about 3000 tracks
Now With Web Data
Now With Web Data

Combining Audio-based Similarity with Web-based Data to Accelerate Automatic Music Playlist Generation
Peter Knees, Tim Pohle, Markus Schedl, and Gerhard Widmer
Now With Web Data

Combining Audio-based Similarity with Web-based Data to Accelerate Automatic Music Playlist Generation

Peter Knees, Tim Pohle, Markus Schedl, and Gerhard Widmer
Graph Methods
Dijkstra's algorithm

1. Assign to every node a distance value. Set it to zero for our initial node and to infinity for all other nodes.
2. Mark all nodes as unvisited. Set initial node as current.
3. For current node, consider all its unvisited neighbors and calculate their tentative distance (from the initial node).
4. When we are done considering all neighbors of the current node, mark it as visited. A visited node will not be checked ever again; its distance recorded now is final and minimal.
5. If all nodes have been visited, finish. Otherwise, set the unvisited node with the smallest distance (from the initial node) as the next "current node" and continue from step 3.
Graph Methods

Dijkstra's algorithm

![Graph with nodes and edges labeled with distances]
Graph Methods

Dijkstra's algorithm

![Graph Diagram]
Graph Methods

Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods

Dijkstra's algorithm
Graph Methods

Dijkstra's algorithm
Graph Methods

Dijkstra's algorithm
Graph Methods

Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm

Out

1

2

3

4

5

6

7

9

10

11

15

14

9

2

6

∞
Graph Methods
Dijkstra's algorithm
Graph Methods

Dijkstra's Algorithm
Graph Methods

Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm

14 > 9 + 2

Out

Out

Out

Out
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
Dijkstra's algorithm
Graph Methods
minimum spanning tree
Graph Methods
min cut/max flow

Social Playlists and Bottleneck Measurements:
Exploiting Musician Social Graphs Using Content-Based Dissimilarity and Pairwise Maximum Flow Values
Fields, Ben and Jacobson, Kurt and Rhodes, Christophe and Casey, Michael
Social Playlists and Bottleneck Measurements: Exploiting Musician Social Graphs Using Content-Based Dissimilarity and Pairwise Maximum Flow Values
Fields, Ben and Jacobson, Kurt and Rhodes, Christophe and Casey, Michael
Graph-Based Path Finding

- A **directed graph** is created based on the **friend** connections amongst artists found on **myspace**

- The edges of this graph are weighted using content-based similarity

- Playlists are constructed through the use of the **max flow/min cut** from a starting to ending artist
Break
Part II
Points-In-Space
Points-In-Space
Points-In-Space
Start-End Timbrel Paths

1. For every song, calculate divergence from select start \( D_{KL}(i, s) \) and end \( D_{KL}(i, e) \) songs

2. Find \( d\% \) songs with highest divergence from start song; repeat against end song. Remove songs that appear in both sets.

3. Compute divergent ratio for remaining songs:

\[
R(i) = \frac{D_{KL}(i, s)}{D_{KL}(i, e)}
\]
Start-End Timbrel Paths

4. Compute ideal step width:

\[ \text{step} = \frac{R(s) - R(e)}{p + 1} \]

5. Generate ideal positions for each song:

\[ \hat{R}(j) = R(s) + j \ast \text{step} \]

6. Select ideal songs that best match the ideal:

\[ S_j = \arg \min_{i=1,...,m} |\hat{R}(j) - R(i)| \]
Evaluating S-E Paths

objective analysis

- The playlist should contain mostly songs from genres A and B

- At the beginning of the playlist, most songs should be from genre A, at the end from genre B and from both genres in the middle
Evaluating S-E Paths

goalie analysis

<table>
<thead>
<tr>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>33</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td>8</td>
</tr>
<tr>
<td>Sec2</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>Sec3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>30</td>
<td>5</td>
<td>2</td>
<td>35</td>
<td>8</td>
</tr>
<tr>
<td>Sec2</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>66</td>
<td>5</td>
</tr>
<tr>
<td>Sec3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>70</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>26</td>
<td>7</td>
<td>2</td>
<td>20</td>
<td>7</td>
</tr>
<tr>
<td>Sec2</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>Sec3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>19</td>
<td>3</td>
<td>8</td>
<td>28</td>
<td>13</td>
</tr>
<tr>
<td>Sec2</td>
<td>17</td>
<td>4</td>
<td>4</td>
<td>20</td>
<td>19</td>
</tr>
<tr>
<td>Sec3</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>22</td>
<td>16</td>
</tr>
</tbody>
</table>
Evaluating S-E Paths
objective analysis

<table>
<thead>
<tr>
<th>Sec1</th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>33</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td>8</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>81</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sec1</th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>26</td>
<td>7</td>
<td>2</td>
<td>20</td>
<td>7</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sec1</th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>19</td>
<td>3</td>
<td>8</td>
<td>28</td>
<td>13</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>3</td>
<td>4</td>
<td>20</td>
<td>19</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>22</td>
<td>16</td>
<td>42</td>
</tr>
</tbody>
</table>
Evaluating S-E Paths

objective analysis

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>33</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td>8</td>
<td>38</td>
</tr>
<tr>
<td>Sec2</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>81</td>
</tr>
<tr>
<td>Sec3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>26</td>
<td>7</td>
<td>2</td>
<td>20</td>
<td>7</td>
<td>38</td>
</tr>
<tr>
<td>Sec2</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>80</td>
</tr>
<tr>
<td>Sec3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>30</td>
<td>5</td>
<td>2</td>
<td>35</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>Sec2</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>66</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>Sec3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>70</td>
<td>4</td>
<td>18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>19</td>
<td>3</td>
<td>8</td>
<td>28</td>
<td>13</td>
<td>29</td>
</tr>
<tr>
<td>Sec2</td>
<td>17</td>
<td>4</td>
<td>4</td>
<td>20</td>
<td>19</td>
<td>36</td>
</tr>
<tr>
<td>Sec3</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>22</td>
<td>16</td>
<td>42</td>
</tr>
</tbody>
</table>
Evaluating S-E Paths
objective analysis

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>33</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td>8</td>
<td>38</td>
</tr>
<tr>
<td>Sec2</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>81</td>
</tr>
<tr>
<td>Sec3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>30</td>
<td>5</td>
<td>2</td>
<td>35</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>Sec2</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>66</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>Sec3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>70</td>
<td>4</td>
<td>18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>26</td>
<td>7</td>
<td>2</td>
<td>20</td>
<td>7</td>
<td>38</td>
</tr>
<tr>
<td>Sec2</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>80</td>
</tr>
<tr>
<td>Sec3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>19</td>
<td>3</td>
<td>8</td>
<td>28</td>
<td>13</td>
<td>29</td>
</tr>
<tr>
<td>Sec2</td>
<td>17</td>
<td>4</td>
<td>4</td>
<td>20</td>
<td>19</td>
<td>36</td>
</tr>
<tr>
<td>Sec3</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>22</td>
<td>16</td>
<td>42</td>
</tr>
</tbody>
</table>
Evaluating S-E Paths
objective analysis

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>33</td>
<td>5</td>
<td>2</td>
<td>15</td>
<td>8</td>
<td>38</td>
</tr>
<tr>
<td>Sec2</td>
<td>5</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>81</td>
</tr>
<tr>
<td>Sec3</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>30</td>
<td>5</td>
<td>2</td>
<td>35</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>Sec2</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>66</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>Sec3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>70</td>
<td>4</td>
<td>18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>26</td>
<td>7</td>
<td>2</td>
<td>20</td>
<td>7</td>
<td>38</td>
</tr>
<tr>
<td>Sec2</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>80</td>
</tr>
<tr>
<td>Sec3</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>HiHo</th>
<th>Regg</th>
<th>Funk</th>
<th>Elec</th>
<th>Pop</th>
<th>Rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sec1</td>
<td>19</td>
<td>3</td>
<td>8</td>
<td>28</td>
<td>13</td>
<td>29</td>
</tr>
<tr>
<td>Sec2</td>
<td>17</td>
<td>4</td>
<td>4</td>
<td>20</td>
<td>19</td>
<td>36</td>
</tr>
<tr>
<td>Sec3</td>
<td>12</td>
<td>3</td>
<td>4</td>
<td>22</td>
<td>16</td>
<td>42</td>
</tr>
</tbody>
</table>
Evaluating S-E Paths
subjective analysis

• How many outliers are in the playlist which do not fit the overall flavor of the playlist?

• Is the order of songs in the playlist from the start to the end song apparent?
Evaluating S-E Paths
subjective analysis

<table>
<thead>
<tr>
<th>Genres from</th>
<th>Genres to</th>
<th># of outliers</th>
<th>order apparent</th>
</tr>
</thead>
<tbody>
<tr>
<td>HiHo</td>
<td>Regg</td>
<td>4.7</td>
<td>x</td>
</tr>
<tr>
<td>HiHo</td>
<td>Funk</td>
<td>1.7</td>
<td>xx</td>
</tr>
<tr>
<td>HiHo</td>
<td>Elec</td>
<td>1.3</td>
<td>x</td>
</tr>
<tr>
<td>HiHo</td>
<td>Pop</td>
<td>2.7</td>
<td>xx</td>
</tr>
<tr>
<td>HiHo</td>
<td>Rock</td>
<td>0</td>
<td>xxx</td>
</tr>
<tr>
<td>Regg</td>
<td>Funk</td>
<td>0.7</td>
<td>xx</td>
</tr>
<tr>
<td>Regg</td>
<td>Elec</td>
<td>1.3</td>
<td>x</td>
</tr>
<tr>
<td>Regg</td>
<td>Pop</td>
<td>1.3</td>
<td>xxx</td>
</tr>
<tr>
<td>Regg</td>
<td>Rock</td>
<td>0.3</td>
<td>xx</td>
</tr>
<tr>
<td>Funk</td>
<td>Elec</td>
<td>1.0</td>
<td>xx</td>
</tr>
<tr>
<td>Funk</td>
<td>Pop</td>
<td>1.7</td>
<td>x</td>
</tr>
<tr>
<td>Funk</td>
<td>Rock</td>
<td>0</td>
<td>xx</td>
</tr>
<tr>
<td>Elec</td>
<td>Pop</td>
<td>0</td>
<td>xxx</td>
</tr>
<tr>
<td>Elec</td>
<td>Rock</td>
<td>0</td>
<td>x</td>
</tr>
<tr>
<td>Pop</td>
<td>Rock</td>
<td>0</td>
<td>xxx</td>
</tr>
<tr>
<td>average</td>
<td></td>
<td>1.1</td>
<td>71.1%</td>
</tr>
</tbody>
</table>

Table 8

Evaluating S-E Paths
subjective analysis

As with the objective evaluation in Sec. 4.1, relaxing the base might not even contain songs fitting in between. An effect at the middle of playlists: the first half would be very close to the start song, the second half to the end song but a sort of smooth transition is missing. This was sometimes very well with it being apparent in certain types of rock songs. Other sources of mistakes are to dominate the models giving rise to high similarities with about order are given.

The sequential order of the playlists seems to work similar to each other no matter what the genres of the songs are. The sequential order of the playlists appears to be very well with it being apparent in another very well with it being apparent in another.

One problem with the average number of outliers in a playlist is quite low, this means that on average, a user will find it easy to see the order.
Evaluating S-E Paths
subjective analysis

<table>
<thead>
<tr>
<th>Genres from</th>
<th>Genres to</th>
<th># of outliers</th>
<th>order apparent</th>
</tr>
</thead>
<tbody>
<tr>
<td>HiHo</td>
<td>Regg</td>
<td>4.7</td>
<td>x</td>
</tr>
<tr>
<td>HiHo</td>
<td>Funk</td>
<td>1.7</td>
<td>xx</td>
</tr>
<tr>
<td>HiHo</td>
<td>Elec</td>
<td>1.3</td>
<td>xxx</td>
</tr>
<tr>
<td>HiHo</td>
<td>Pop</td>
<td>2.7</td>
<td>xx</td>
</tr>
<tr>
<td>HiHo</td>
<td>Rock</td>
<td>0</td>
<td>xxx</td>
</tr>
<tr>
<td>Regg</td>
<td>Funk</td>
<td>0.7</td>
<td>xx</td>
</tr>
<tr>
<td>Regg</td>
<td>Elec</td>
<td>1.3</td>
<td>xxx</td>
</tr>
<tr>
<td>Regg</td>
<td>Pop</td>
<td>1.3</td>
<td>xxx</td>
</tr>
<tr>
<td>Regg</td>
<td>Rock</td>
<td>0.3</td>
<td>xx</td>
</tr>
<tr>
<td>Funk</td>
<td>Elec</td>
<td>1.0</td>
<td>xx</td>
</tr>
<tr>
<td>Funk</td>
<td>Pop</td>
<td>1.7</td>
<td>xx</td>
</tr>
<tr>
<td>Funk</td>
<td>Rock</td>
<td>0</td>
<td>xx</td>
</tr>
<tr>
<td>Elec</td>
<td>Pop</td>
<td>0</td>
<td>xxx</td>
</tr>
<tr>
<td>Elec</td>
<td>Rock</td>
<td>0</td>
<td>xx</td>
</tr>
<tr>
<td>Pop</td>
<td>Rock</td>
<td>0</td>
<td>xxx</td>
</tr>
</tbody>
</table>

average 1.1 71.1% 17.8% 11.1%
Evaluating S-E Paths
subjective analysis

<table>
<thead>
<tr>
<th>Genres from</th>
<th>Genres to</th>
<th># of outliers</th>
<th>order apparent</th>
</tr>
</thead>
<tbody>
<tr>
<td>HiHo</td>
<td>Regg</td>
<td>4.7</td>
<td>x</td>
</tr>
<tr>
<td>HiHo</td>
<td>Funk</td>
<td>1.7</td>
<td>xx</td>
</tr>
<tr>
<td>HiHo</td>
<td>Elec</td>
<td>1.3</td>
<td>xxx</td>
</tr>
<tr>
<td>HiHo</td>
<td>Pop</td>
<td>2.7</td>
<td>xx</td>
</tr>
<tr>
<td>HiHo</td>
<td>Rock</td>
<td>0</td>
<td>xxx</td>
</tr>
<tr>
<td>Regg</td>
<td>Funk</td>
<td>0.7</td>
<td>xx</td>
</tr>
<tr>
<td>Regg</td>
<td>Elec</td>
<td>1.3</td>
<td>xxx</td>
</tr>
<tr>
<td>Regg</td>
<td>Pop</td>
<td>1.3</td>
<td>xxx</td>
</tr>
<tr>
<td>Regg</td>
<td>Rock</td>
<td>0.3</td>
<td>xx</td>
</tr>
<tr>
<td>Funk</td>
<td>Elec</td>
<td>1.0</td>
<td>xx</td>
</tr>
<tr>
<td>Funk</td>
<td>Pop</td>
<td>1.7</td>
<td>xx</td>
</tr>
<tr>
<td>Funk</td>
<td>Rock</td>
<td>0</td>
<td>xx</td>
</tr>
<tr>
<td>Elec</td>
<td>Pop</td>
<td>0</td>
<td>xxx</td>
</tr>
<tr>
<td>Elec</td>
<td>Rock</td>
<td>0</td>
<td>xx</td>
</tr>
<tr>
<td>Pop</td>
<td>Rock</td>
<td>0</td>
<td>xxx</td>
</tr>
<tr>
<td>average</td>
<td></td>
<td>1.1</td>
<td>71.1%</td>
</tr>
</tbody>
</table>
Playlist Similarity

• The co-occurrence of objects in an authored stream can be used as a proxy for object similarity

• This sort of similarity is especially effective for the generation of playlists

• Employs the use of an undirected graph, weighted by co-occurrence counts
Playlist Similarity

Inferring similarity between music objects with application to playlist generation
R. Ragno and C.J.C. Burges and C. Herley
inferring similarity between music objects with application to playlist generation

R. Ragno and C.J.C. Burges and C. Herley

4. EXPERIMENTS

4.1 Examples Playlists

We now present a few sample playlists to illustrate

<table>
<thead>
<tr>
<th>Track</th>
<th>Artist</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paperback Writer</td>
<td>Beatles</td>
<td>0.0</td>
</tr>
<tr>
<td>Breakfast In America</td>
<td>Supertramp</td>
<td>8.607</td>
</tr>
<tr>
<td>We’re An American Band</td>
<td>Grand Funk Railroad</td>
<td>8.607</td>
</tr>
<tr>
<td>In The Dark</td>
<td>Billy Squier</td>
<td>17.244</td>
</tr>
<tr>
<td>I Shot The Sheriff</td>
<td>Eric Clapton</td>
<td>12.192</td>
</tr>
<tr>
<td>Fat Bottomed Girls</td>
<td>Queen</td>
<td>16.335</td>
</tr>
<tr>
<td>Jumpin’ Jack Flash</td>
<td>The Rolling Stones</td>
<td>13.723</td>
</tr>
<tr>
<td>Working For The Weekend</td>
<td>Loverboy</td>
<td>15.251</td>
</tr>
<tr>
<td>Dream Weaver</td>
<td>Gary Wright</td>
<td>15.520</td>
</tr>
<tr>
<td>Smells Like Teen Spirit!</td>
<td>Nirvana</td>
<td>15.735</td>
</tr>
<tr>
<td>Can’t Stop</td>
<td>Red Hot Chili Peppers</td>
<td>16.732</td>
</tr>
<tr>
<td>Still Waiting</td>
<td>Sum 41</td>
<td>19.256</td>
</tr>
<tr>
<td>Grave Digger</td>
<td>Dave Matthews</td>
<td>20.665</td>
</tr>
</tbody>
</table>

Lithium [Nirvana] : 0.0

<table>
<thead>
<tr>
<th>Track</th>
<th>Artist</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall To Pieces</td>
<td>Velvet Revolver</td>
<td>7.668</td>
</tr>
<tr>
<td>Tonight, Tonight</td>
<td>Smashing Pumpkins</td>
<td>12.712</td>
</tr>
<tr>
<td>Slow Hands</td>
<td>Interpol</td>
<td>12.712</td>
</tr>
<tr>
<td>Renegades Of Funk</td>
<td>Rage Against The Machine</td>
<td>10.127</td>
</tr>
<tr>
<td>Before I Forget</td>
<td>Slipknot</td>
<td>7.355</td>
</tr>
<tr>
<td>The Kids Aren’t Alright</td>
<td>Offspring</td>
<td>11.712</td>
</tr>
<tr>
<td>All These Things That I’ve Done</td>
<td>The Killers</td>
<td>9.542</td>
</tr>
<tr>
<td>Weapon</td>
<td>Matthew Good</td>
<td>18.914</td>
</tr>
<tr>
<td>Kryptonite</td>
<td>3 Doors Down</td>
<td>11.127</td>
</tr>
<tr>
<td>Home</td>
<td>Three Days Grace</td>
<td>8.712</td>
</tr>
<tr>
<td>Whatever</td>
<td>Godsmack</td>
<td>10.127</td>
</tr>
<tr>
<td>Colors</td>
<td>Crossfade</td>
<td>7.097</td>
</tr>
</tbody>
</table>

Inferring similarity between music objects with application to playlist generation

R. Ragno and C.J.C. Burges and C. Herley
Playlist Similarity

example similarities

<table>
<thead>
<tr>
<th>Hey Jude [Beatles]</th>
<th>0.000</th>
<th>Highway To Hell [AC/DC]</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Saturday In The Park [Chicago]</td>
<td>8.000</td>
<td>Be Yourself [Audioslave]</td>
<td>6.558</td>
</tr>
<tr>
<td>Shine It All Around [Robert Plant]</td>
<td>8.000</td>
<td>The Hand That Feeds [Nine Inch Nails]</td>
<td>6.584</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shine It All Around [Robert Plant]</td>
<td>6.982</td>
</tr>
</tbody>
</table>
Playlist Steering

- Create a timbrel features
- Create the space using tuple and triple n-gram sequences from playlist logs
- Generate playlists via tag steering
Playlist Steering

1. Select a seed track

2. Threshold transition matrix to generate set of possible next tracks

3. User creates a tag cloud, assigning weights to any of 360 tags

4. Autotagger creates tag cloud for all candidate tracks selected in (2). Cosine distance is taken between the user’s tag cloud and each song’s.

5. The track with the minimum cosine distance from seed is played
Playlist Steering

<table>
<thead>
<tr>
<th>Soft tag cloud</th>
<th>Hard tag cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viva la Vida by Coldplay</td>
<td>All I Want by Staind</td>
</tr>
<tr>
<td>Wish You Were Here by Pink Floyd</td>
<td>Re-Education (Through Labor) by Rise Against</td>
</tr>
<tr>
<td>Peaceful, Easy Feeling by Eagles</td>
<td>Hammerhead by The Offspring</td>
</tr>
<tr>
<td>With or Without You by U2</td>
<td>The Kill by 30 Seconds To Mars</td>
</tr>
<tr>
<td>One by U2</td>
<td>When You Were Young by The Killers</td>
</tr>
<tr>
<td>Fields Of Gold by Sting</td>
<td>Hypnotize by System of a Down</td>
</tr>
<tr>
<td>Every Breath You Take by The Police</td>
<td>Breath by Breaking Benjamin</td>
</tr>
<tr>
<td>Gold Dust Woman by Fleetwood Mac</td>
<td>My Hero by Foo Fighters</td>
</tr>
<tr>
<td>Enjoy The Silence by Depeche Mode</td>
<td>Turn The Page by Metallica</td>
</tr>
</tbody>
</table>
Playlist Steering
Scaling up playlisting
Scaling up playlist generation

• Building playlists involves satisfying constraints.
• Global constraints: no duplicate songs, tempo between 120 and 130 BPM
• Ordering constraints: no consecutive artists, DMCA rules
• Sorting constraints: ordered by danceability and loudness
• Playlist length: 15 songs, 32 minutes, < 20mb
• Finite constraint satisfaction problem. It’s NP-HARD
General Approach

- Playlist is a sequence of songs: \( S_1, S_2 \ldots S_n \) drawn from a large pool of songs
- \( \text{Cost}(S_n, C) \) is how well song S at position N satisfies constraint C
- \( \text{Cost}(S_n) \) is total cost for song S at position N for all constraints
- \( \text{Cost}(P) \) is total cost of all songs in the Playlist
- Goal: Find \( S_1, \ldots, S_n \) that minimizes \( \text{Cost}(P) \)
Scaling up playlist generation

Generate random playlist

while Cost(P) > threshold:
    Calculate Cost(Sn) for each song
    find max( Cost(Sn) ) that is not tabu
    find best possible replacement

worst variables for which no value can be found to decrease the total cost are labelled as **tabu** for a given number of iterations.

Typical runtime: 1.4 seconds for 10 song playlist from a pool of 20,000 songs with 10 constraints
Fast Generation of Optimal Music Playlists using Local Search

- Simulated annealing
- Heuristic improvements
- Song domain reduction
- Two level search:
  1. Replace, Insert, Delete
  2. Swap
- Partial constraint voting

```latex
\textbf{INITIALIZE} \ p, \ t_0, \ L_0;
\ h := 0;
\ r := 0;
\textbf{repeat}
\quad \textbf{for} \ l := 1 \ \textbf{to} \ L_h \ \textbf{do}
\quad \begin{align*}
& \textbf{if} \ r < \beta \ \textbf{then} \\
& \quad \begin{align*}
& \textbf{if} \ \delta > \text{random}[0,1] \ \textbf{then} \\
& \quad \text{GENERATE} \ \text{RANDOM} \ p' \in N_{\text{reselect}}(p) \\
& \quad \textbf{else} \\
& \quad \text{GENERATE} \ p' \in N_{\text{reselect}}(p) \ \text{BY VOTING;} \\
& \quad \textbf{if} \ f(p') \leq f(p) \ \textbf{or} \ \exp(\frac{f(p)-f(p')}{t}) > \text{random}[0,1] \\
& \quad \textbf{then} \ p := p'; \\
& \quad r := r + 1
\end{align*}
\end{align*}
\textbf{else}
\quad \begin{align*}
& p := NDR(p, \gamma); \\
& r := 0
\end{align*}
\quad \textbf{end}
\quad \textbf{end}
\quad h := h + 1;
\quad \text{CALCULATE LENGTH} \ L_h;
\quad \text{CALCULATE CONTROL} \ t_h
\\textbf{until} \ \text{STOP CRITERION}
```
Fast Generation of Optimal Music Playlists using Local Search

- Start at high temperature
- Repeat until stop criterion:
  - Alternate - 100 at a time:
    - Level 1:
      - Select *Replace*, *Insert* or *Delete* operation by *voting* or *random*
      - Create new playlist by applying the operation
      - Accept the change if:
        - It lowers the overall cost or
        - Randomly when at high temperatures
      - Lower the temperature
    - Level 2:
      - Non-deteriorating *swap*

Typical runtime: 2 seconds for 14 song playlist with 15 constraints from a pool of 2,000 songs
The Echo Nest playlister

- Start with millions of songs
- Apply global constraints to create smaller song pool (1K to 10K songs)
- Use constraint engine to find best playlist:
  - Beam search
  - Adaptive search
  - Populate with data
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Beam Search
Group Playlisting

• Group playlisting:
  • Radio, Clubs, Offices, Health Clubs, The Web

• Group playlisting challenges:
  • Varying and conflicting music tastes
  • Different levels of assertiveness

• Traditional:
  • Dictator, Compromise
  • Random, Opt-out
Group cost functions

• New cost functions for group playlisting - **social cost function**:
  • Average happiness - group vote of members
  • Maximum happiness - vote of the happiest group member
  • Minimum misery - vote of the least happy
## Group costs

<table>
<thead>
<tr>
<th></th>
<th>Ben</th>
<th>Paul</th>
<th>Oscar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Max</td>
<td>6</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Min</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Ben</td>
<td>Paul</td>
<td>Oscar</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>1st</td>
<td>2</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>2nd</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3rd</td>
<td>6</td>
<td>2</td>
<td>7</td>
</tr>
</tbody>
</table>
## Group costs

<table>
<thead>
<tr>
<th></th>
<th>Ben</th>
<th>Paul</th>
<th>Oscar</th>
<th>Avg</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>10</td>
<td>1</td>
<td>4.33</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3.33</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
## Group costs

<table>
<thead>
<tr>
<th></th>
<th>Ben</th>
<th>Paul</th>
<th>Oscar</th>
<th>Avg</th>
<th>Max</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>2</td>
<td>10</td>
<td>1</td>
<td>4.33</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>2nd</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3.33</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3rd</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>
Flytrap

- Uses simple voting mechanism - ‘average happiness’
  - Each listener agent votes:
    - Artist previously listened == high votes
    - Genre previous listened == positive vote
    - Songs with more votes have higher probability of being played
    - Never play 2 songs by same artist in a row
    - Loose coherence of genre across tracks
1. Translate the request histories of all requesters into ratings for artists.

2. Predict ratings for each artist that a requester has never requested.

3. Determine what artists are the most popular among the listening audience.

4. Determine what artists are similar to the final artist on the playlist.

5. Select a song to play that is performed by an artist that is both popular among the listening requesters and similar to the artist that precedes it.
How to Combine Different Individual Preferences

The goal of the Reuse Process is to combine different individual preferences into a global group ranking of the candidate songs.

Ex.: three listeners have diverging individual preferences over which candidate song to play after I Spy (Pulp):

- **I Spy** (Pulp) retrieved candidates
  - Lazy (Suede): 0.9
  - Go (Moby): 0
  - Uno (Muse): -0.7
  - Drive (R.E.M.): 0.2

- Preferences:?
  - Win.
  - Win.
  - Win.
  - Win.
1. To **avoid misery**, any candidate song that is **hated** by some listener automatically gets the lowest group preference degree.
2. To ensure **fairness**, the group preference degree of the remaining candidates equals to the **average** of the individual preferences.

How to Combine Different Individual Preferences

- **I Spy (Pulp)**
- **Lazy (Suede)**
- **Go (Moby)**
- **Uno (Muse)**
- **Drive (R.E.M.)**
3. To guarantee individual satisfactions, listeners whose preferred song was not selected in this turn are to be favoured next.
4. The **satisfaction degree** of a listener for previous songs changes her **weight** in the calculation of the average group preference.

<table>
<thead>
<tr>
<th>Song</th>
<th>Weight</th>
<th>Score</th>
<th>Preference</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lazy (Suede)</td>
<td>-0.2</td>
<td>1</td>
<td>-0.2</td>
<td>not satisfied</td>
</tr>
<tr>
<td>Loser (Beck)</td>
<td>0.2</td>
<td>0</td>
<td>0.2</td>
<td>satisfied</td>
</tr>
<tr>
<td>Song 2 (Blur)</td>
<td>0.2</td>
<td>-0.3</td>
<td>-0.6</td>
<td>not satisfied</td>
</tr>
<tr>
<td>Flower (Eels)</td>
<td>0.2</td>
<td>0.8</td>
<td>0.16</td>
<td>satisfied</td>
</tr>
<tr>
<td>Joga (Björk)</td>
<td>0.2</td>
<td>0.6</td>
<td>0.12</td>
<td>satisfied</td>
</tr>
</tbody>
</table>

The **satisfaction degree** changes the weight in the calculation of the average group preference.
Beat-matching
Crossfading
Beat-matching and crossfading

- Select songs with similar tempos
- Select transition location
  - Similar rhythmic pattern
- Specific sections (last 30 seconds of song 1 and first 30 seconds of song 2)
- Align their beats over the course of a transition
- Crossfade the volumes
First, find the beats
Figure 6-1: Time-scaling example of a typical sound segment. Note that we only process the decay part of the sound. The energy is preserved by overlapping and adding Hanning windows by 50%. In this example we stretch the whole segment [top] by an additional 30% [bottom].
Beat-matching and crossfading

Creating Music by Listening
by Tristan Jehan
Some Examples
Some Examples

Rihanna (122 bpm)

Gotan Project (95 bpm)
Some Examples

Rihanna (122 bpm) (95 bpm) Gotan Project
Some Examples

Bob Marley to Bob Marley

Rihanna (122 bpm) → (95 bpm) Gotan Project
Some Examples

Bob Marley to Bob Marley

Rihanna (122 bpm) to Gotan Project (95 bpm)
Some Examples

Bob Marley to Bob Marley

Rihanna (122 bpm) to Gotan Project (95 bpm)
Some Examples

Bob Marley to Bob Marley
Sade to Sting

Rihanna (122 bpm)  Gotan Project (95 bpm)
Some Examples

Bob Marley to Bob Marley
Sade to Sting

Rihanna (122 bpm) to Gotan Project (95 bpm)
Some Examples

Bob Marley to Bob Marley
Sade to Sting

Rihanna (122 bpm)  
(95 bpm) Gotan Project
Some Examples

Bob Marley to Bob Marley
Sade to Sting
April March to April March

Rihanna (122 bpm)                          Gotan Project (95 bpm)
Some Examples

Bob Marley to Bob Marley
Sade to Sting
April March to April March
Evaluating playlists
Subjective Analysis
Direct Listening Tests hypotheses

1. Playlists compiled by PATS contain more preferred songs than randomly assembled playlists, irrespective of a given context-of-use.

2. Similarly, PATS playlists are rated higher than randomly assembled playlists, irrespective of a given context-of-use.
Direct Listening Tests hypotheses

3. Successive playlists compiled by PATS contain an increasing number of preferred songs.

4. Similarly, successive PATS playlists are successively rated higher.

5. Successive playlists compiled by PATS contain more distinct and preferred songs than randomly assembled playlists.
Direct Listening Tests
set-up

• Three measures: **precision, coverage** and **rating score**

• 20 participants (17m, 3f), 8 sessions over 4 days per participant
  • User selects a song, given a context (4 playlist per context)
  • A PATS playlist and a random playlist are generated (11 songs each, 1 minute excerpts)
  • Judgements expressed per song, ratings per playlist
Direct Listening Tests results

![Graphs showing mean precision for PATS and random in sessions for soft and lively music.](Image)
Direct Listening Tests results

- **Graph a. Soft music**
  - PATS
  - Random

- **Graph b. Lively music**
  - PATS
  - Random

PATS: Realization and User Evaluation of an Automatic Playlist Generator
Steffen Pauws and Berry Eggen
Direct Listening Tests results

![Graph of mean rating score for soft music and lively music with PATS and random sessions.](image)
Skip-Based Listening Tests basics

- Evaluation integrated into system
- Assumptions:
  1. A seed song is given
  2. A skip button is available and easily accessible to the user
  3. A lazy user who is willing to sacrifice quality for time
Skip-Based Listening Tests

use cases

1. The user wants to listen to songs that are similar to the seed song

2. Same as (1) but with a dislike of an arbitrary artist for a subjective reason (e.g., taste)

3. The user’s preference changes over time. Specifically, in a 20 song playlist, the first 5 songs from genre A, the middle 10 from either genre A or B, last 5 songs from genre B.
Skip-Based Listening Tests

heuristics

A. $N$ nearest neighbors to the seed song are played ($N = \text{accepted} + \text{skipped}$). This heuristic is the baseline.

B. The candidate song closest to the last song accepted by the user is played. This is like (A) except the seed song is always the last song accepted.

C. The candidate song closest to any of the accepted songs is played.

D. For each candidate song, let $da$ be the distance to the nearest accepted, and let $ds$ be the distance to the nearest skipped. If $da < ds$, then add the candidate to the set $S$. From $S$ play the song with smallest $da$. If $S$ is empty, then play the candidate song which has the best (i.e. the lowest) $da/ds$ ratio.
Skip-Based Listening Tests
skips in UCI

<table>
<thead>
<tr>
<th>Genres</th>
<th>Artists</th>
<th>Tracks</th>
<th>Artists/Genre Min</th>
<th>Artists/Genre Max</th>
<th>Tracks/Genre Min</th>
<th>Tracks/Genre Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>103</td>
<td>2522</td>
<td>3</td>
<td>6</td>
<td>45</td>
<td>259</td>
</tr>
</tbody>
</table>
Skip-Based Listening Tests

skips in UC1

(a) Heuristic A

(b) Heuristic D

Figure 1: Skips per playlist position for UC-1.

Table 2: Number of skips for UC-1 and UC-2.
Skip-Based Listening Tests

UC1 and UC2 skips

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Min</th>
<th>Median</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>37.0</td>
<td>133.0</td>
<td>2053</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>30.0</td>
<td>164.4</td>
<td>2152</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>14.0</td>
<td>91.0</td>
<td>1298</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>11.0</td>
<td>23.9</td>
<td>425</td>
</tr>
</tbody>
</table>

| UC-2      |     |        |      |     |
| A         | 0   | 52.0   | 174.0| 2230|
| B         | 0   | 36.0   | 241.1| 2502|
| C         | 0   | 17.0   | 116.9| 1661|
| D         | 0   | 15.0   | 32.9 | 453 |
### Skip-Based Listening Tests

#### UC1 and UC2 skips

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Min</th>
<th>Median</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UC-1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>37.0</td>
<td>133.0</td>
<td>2053</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>30.0</td>
<td>164.4</td>
<td>2152</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>14.0</td>
<td>91.0</td>
<td>1298</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>11.0</td>
<td>23.9</td>
<td>425</td>
</tr>
<tr>
<td><strong>UC-2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>52.0</td>
<td>174.0</td>
<td>2230</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>36.0</td>
<td>241.1</td>
<td>2502</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>17.0</td>
<td>116.9</td>
<td>1661</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>15.0</td>
<td>32.9</td>
<td>453</td>
</tr>
</tbody>
</table>
Skip-Based Listening Tests
UC1 and UC2 skips

<table>
<thead>
<tr>
<th>Heuristic</th>
<th>Min</th>
<th>Median</th>
<th>Mean</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>37.0</td>
<td>133.0</td>
<td>2053</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>30.0</td>
<td>164.4</td>
<td>2152</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>14.0</td>
<td>91.0</td>
<td>1298</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>11.0</td>
<td>23.9</td>
<td>425</td>
</tr>
<tr>
<td>UC-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>52.0</td>
<td>174.0</td>
<td>2230</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>36.0</td>
<td>241.1</td>
<td>2502</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>17.0</td>
<td>116.9</td>
<td>1661</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>15.0</td>
<td>32.9</td>
<td>453</td>
</tr>
</tbody>
</table>
Dynamic Playlist Generation Based on Skipping Behavior

Elias Pampalk and T. Pohle and G. Widmer

Skip-Based Listening Tests

<table>
<thead>
<tr>
<th>UC3 skips</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Start</th>
<th>Goto</th>
<th>Heuristic A</th>
<th>Heuristic B</th>
<th>Heuristic C</th>
<th>Heuristic D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Median</td>
<td>Mean</td>
<td>Median</td>
<td>Mean</td>
</tr>
<tr>
<td>Euro-Dance</td>
<td>Trance</td>
<td>69.0</td>
<td>171.4</td>
<td>36.0</td>
<td>64.9</td>
</tr>
<tr>
<td>Trance</td>
<td>Euro-Dance</td>
<td>66.0</td>
<td>149.1</td>
<td>24.0</td>
<td>79.1</td>
</tr>
<tr>
<td>German Hip Hop</td>
<td>Hard Core Rap</td>
<td>33.0</td>
<td>61.9</td>
<td>32.0</td>
<td>45.6</td>
</tr>
<tr>
<td>Hard Core Rap</td>
<td>German Hip Hop</td>
<td>21.5</td>
<td>32.2</td>
<td>18.0</td>
<td>51.9</td>
</tr>
<tr>
<td>Heavy Metal/Thrash</td>
<td>Death Metal</td>
<td>98.5</td>
<td>146.4</td>
<td>54.0</td>
<td>92.5</td>
</tr>
<tr>
<td>Death Metal</td>
<td>Heavy Metal/Thrash</td>
<td>14.0</td>
<td>69.2</td>
<td>16.0</td>
<td>53.7</td>
</tr>
<tr>
<td>Bossa Nova</td>
<td>Jazz Guitar</td>
<td>68.5</td>
<td>228.1</td>
<td>32.0</td>
<td>118.7</td>
</tr>
<tr>
<td>Jazz Guitar</td>
<td>Bossa Nova</td>
<td>21.0</td>
<td>26.7</td>
<td>22.0</td>
<td>21.5</td>
</tr>
<tr>
<td>Jazz Guitar</td>
<td>Jazz</td>
<td>116.0</td>
<td>111.3</td>
<td>53.0</td>
<td>75.7</td>
</tr>
<tr>
<td>Jazz</td>
<td>Jazz Guitar</td>
<td>512.5</td>
<td>717.0</td>
<td>1286.0</td>
<td>1279.5</td>
</tr>
<tr>
<td>A Cappella</td>
<td>Death Metal</td>
<td>1235.0</td>
<td>1230.5</td>
<td>1523.0</td>
<td>1509.9</td>
</tr>
<tr>
<td>Death Metal</td>
<td>A Cappella</td>
<td>1688.0</td>
<td>1647.2</td>
<td>1696.0</td>
<td>1653.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Acknowledgements</th>
</tr>
</thead>
</table>

This research was supported by the EU project FP6-507142 (based on data from long term usage) are other options. Similarity or modeling the user's context more accurately might be to track the direction of this change. Incorporating additional information such as web-based artist similarity or listening history could improve accuracy. For use cases related to changing user preferences a key measure. Any improvements would lead to fewer skips.

Dynamic Playlist Generation Based on Skipping Behavior

Elias Pampalk and T. Pohle and G. Widmer

162
## Skip-Based Listening Tests

### UC3 skips

<table>
<thead>
<tr>
<th>Start</th>
<th>Goto</th>
<th>Heuristic A</th>
<th>Heuristic B</th>
<th>Heuristic C</th>
<th>Heuristic D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Median</td>
<td>Mean</td>
<td>Median</td>
<td>Mean</td>
</tr>
<tr>
<td>Euro-Dance</td>
<td>Trance</td>
<td>69.0</td>
<td>171.4</td>
<td>36.0</td>
<td>64.9</td>
</tr>
<tr>
<td></td>
<td>Euro-Dance</td>
<td>66.0</td>
<td>149.1</td>
<td>24.0</td>
<td>79.1</td>
</tr>
<tr>
<td>Trance</td>
<td>German Hip Hop</td>
<td>33.0</td>
<td>61.9</td>
<td>32.0</td>
<td>45.6</td>
</tr>
<tr>
<td></td>
<td>Hard Core Rap</td>
<td>21.5</td>
<td>32.2</td>
<td>18.0</td>
<td>51.9</td>
</tr>
<tr>
<td>German Hip Hop</td>
<td>Hard Core Rap</td>
<td>98.5</td>
<td>146.4</td>
<td>54.0</td>
<td>92.5</td>
</tr>
<tr>
<td></td>
<td>Death Metal</td>
<td>14.0</td>
<td>69.2</td>
<td>16.0</td>
<td>53.7</td>
</tr>
<tr>
<td>Death Metal</td>
<td>Heavy Metal/Thrash</td>
<td>68.5</td>
<td>228.1</td>
<td>32.0</td>
<td>118.7</td>
</tr>
<tr>
<td></td>
<td>Jazz Guitar</td>
<td>21.0</td>
<td>26.7</td>
<td>22.0</td>
<td>21.5</td>
</tr>
<tr>
<td>Bossa Nova</td>
<td>Jazz</td>
<td>116.0</td>
<td>111.3</td>
<td>53.0</td>
<td>75.7</td>
</tr>
<tr>
<td>Jazz Guitar</td>
<td>Bossa Nova</td>
<td>512.5</td>
<td>717.0</td>
<td>1286.0</td>
<td>1279.5</td>
</tr>
<tr>
<td>Jazz</td>
<td>Jazz Guitar</td>
<td>1235.0</td>
<td>1230.5</td>
<td>1523.0</td>
<td>1509.9</td>
</tr>
<tr>
<td>A Cappella</td>
<td>Death Metal</td>
<td>1688.0</td>
<td>1647.2</td>
<td>1696.0</td>
<td>1653.9</td>
</tr>
</tbody>
</table>
## Skip-Based Listening Tests

### UC3 skips

<table>
<thead>
<tr>
<th>Start</th>
<th>Goto</th>
<th>Heuristic A</th>
<th>Heuristic B</th>
<th>Heuristic C</th>
<th>Heuristic D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Median</td>
<td>Mean</td>
<td>Median</td>
<td>Mean</td>
</tr>
<tr>
<td>Euro-Dance</td>
<td>Trance</td>
<td>69.0</td>
<td>171.4</td>
<td>36.0</td>
<td>64.9</td>
</tr>
<tr>
<td>Trance</td>
<td>Euro-Dance</td>
<td>66.0</td>
<td>149.1</td>
<td>24.0</td>
<td>79.1</td>
</tr>
<tr>
<td>German Hip Hop</td>
<td>Hard Core Rap</td>
<td>33.0</td>
<td>61.9</td>
<td>32.0</td>
<td>45.6</td>
</tr>
<tr>
<td>Hard Core Rap</td>
<td>German Hip Hop</td>
<td>21.5</td>
<td>32.2</td>
<td>18.0</td>
<td>51.9</td>
</tr>
<tr>
<td>Heavy Metal/Thrash</td>
<td>Death Metal</td>
<td>98.5</td>
<td>146.4</td>
<td>54.0</td>
<td>92.5</td>
</tr>
<tr>
<td>Death Metal</td>
<td>Heavy Metal/Thrash</td>
<td>14.0</td>
<td>69.2</td>
<td>16.0</td>
<td>53.7</td>
</tr>
<tr>
<td>Bossa Nova</td>
<td>Jazz Guitar</td>
<td>68.5</td>
<td>228.1</td>
<td>32.0</td>
<td>118.7</td>
</tr>
<tr>
<td>Jazz Guitar</td>
<td>Bossa Nova</td>
<td>21.0</td>
<td>26.7</td>
<td>22.0</td>
<td>21.5</td>
</tr>
<tr>
<td>Jazz Guitar</td>
<td>Jazz</td>
<td>116.0</td>
<td>111.3</td>
<td>53.0</td>
<td>75.7</td>
</tr>
<tr>
<td>Jazz</td>
<td>Jazz Guitar</td>
<td>512.5</td>
<td>717.0</td>
<td>1286.0</td>
<td>1279.5</td>
</tr>
<tr>
<td>A Cappella</td>
<td>Death Metal</td>
<td>1235.0</td>
<td>1230.5</td>
<td>1523.0</td>
<td>1509.9</td>
</tr>
<tr>
<td>Death Metal</td>
<td>A Cappella</td>
<td>1688.0</td>
<td>1647.2</td>
<td>1696.0</td>
<td>1653.9</td>
</tr>
</tbody>
</table>
Dynamic Heuristics

- Last.fm Radio logs are used to analyze and evaluate several heuristics for dynamic playlists.

- This is done through the treatment of playlists as fuzzy sets.

- Work shows that one heuristic works best given inconsistent rejects while another performs best given inconsistent accepts and third performs equally in either environment.
Dynamic Heuristics

(a) dataset 1

(b) dataset 3

(c) dataset 5

(d) dataset 7

(e) dataset 9
Dynamic Heuristics

Figure 7

(a) $I_{SM}$

(b) $I_{SP}$

(c) $I_{SL} = I_{TL}$

(d) $I_{TP}$

(e) $I_{TM}$
objective analysis
Measuring Distance

We can measure the distance between sequences of tracks using the same methods we can use to measure the distance between frames within tracks.
Measuring Distance

- Topic Modeled Tag Clouds used as a song-level feature
- Sequences of these low dimensional features can then be compared
- The fitness of this pseudo-metric space is examined through patterns in radio playlist logs
Measuring Distance

- Gather tags for all songs
- Create LDA model describing topic distributions
- Infer topic mixtures for all songs
- Create vector database of playlists
Measuring Distance
An evaluation of various playlisting services
Radio Paradise

RP is a blend of many styles and genres of music, carefully selected and mixed by two real human beings. You'll hear modern and classic rock, world music, electronica, even a bit of classical and jazz. What you won't hear are random computer-generated playlists or mind-numbing commercials.

Our specialty is taking a diverse assortment of songs and making them flow together in a way that makes sense harmonically, rhythmically, and lyrically — an art that, to us, is the very essence of radio.

- Listener supported, Internet streaming radio
- 75,000 registered listeners
- Real, radio DJs

radioparadise.com
Playlist Turing Test

The Playlist Survey

In this survey we are looking at the quality of playlists generated by human experts, computer algorithms and random number generators. You will be presented with 12 playlists. For each playlist, we ask you to rate the overall quality of the playlist as a whole and to predict whether the playlist was generated by a professional DJ, a computer algorithm or was created at random. At the end of the survey you'll be shown how accurate your predictions were. More info about this survey can be found at The Playlist Survey

What is your age? 
What is your gender? not specified
How often do you listen to Radio Paradise? not specified

Take the Survey

Expert playlist data graciously provided by Radio Paradise

380 respondents
Radio Paradise Dataset

Playlists collected from Jan 2007 to July 2008

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Playlists</td>
<td>45,283</td>
</tr>
<tr>
<td>Tracks</td>
<td>6,325</td>
</tr>
<tr>
<td>Albums</td>
<td>4,094</td>
</tr>
<tr>
<td>Artists</td>
<td>1971</td>
</tr>
<tr>
<td>Average Length</td>
<td>4.3</td>
</tr>
</tbody>
</table>
### Playlist Turing Test

## The Playlist Survey

**12 Playlists remaining**

<table>
<thead>
<tr>
<th>Image</th>
<th>Playlist Title</th>
<th>Artist</th>
<th>Action 1</th>
<th>Action 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Eels Cover" /></td>
<td>Fresh Feeling</td>
<td>Eels</td>
<td>Play on Spotify</td>
<td>Play in youtube</td>
</tr>
<tr>
<td><img src="image2.png" alt="Flaming Lips Cover" /></td>
<td>Are You A Hypnotist??</td>
<td>Flaming Lips</td>
<td>Play on Spotify</td>
<td>Play in youtube</td>
</tr>
<tr>
<td><img src="image3.png" alt="Forgotten Arm Cover" /></td>
<td>Video</td>
<td>Aimee Mann</td>
<td>Play in youtube</td>
<td></td>
</tr>
<tr>
<td><img src="image4.png" alt="Röyksopp Cover" /></td>
<td>Poor Leno</td>
<td>Röyksopp</td>
<td>Play on Spotify</td>
<td>Play in youtube</td>
</tr>
</tbody>
</table>

**Playlist rating?** [not specified]  **Playlist type?** [not specified]  **Submit**
## Survey Ratings for guesses

<table>
<thead>
<tr>
<th>Playlist guess</th>
<th>Playlist Rating</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Expert</td>
<td>3.33</td>
<td>368</td>
</tr>
<tr>
<td>Algorithm</td>
<td>2.76</td>
<td>373</td>
</tr>
<tr>
<td>Random</td>
<td>2.08</td>
<td>343</td>
</tr>
</tbody>
</table>
### The Survey

#### Survey Ratings for truth

<table>
<thead>
<tr>
<th>Playlist type</th>
<th>Playlist Rating</th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Expert</td>
<td>2.49</td>
<td>400</td>
</tr>
<tr>
<td>Algorithm</td>
<td>2.63</td>
<td>403</td>
</tr>
<tr>
<td>Random</td>
<td>2.64</td>
<td>386</td>
</tr>
</tbody>
</table>
## The Survey

### Confusion Matrix

<table>
<thead>
<tr>
<th>Truth</th>
<th>Human Expert</th>
<th>Algorithm</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Human Expert</strong></td>
<td>121</td>
<td>124</td>
<td>112</td>
</tr>
<tr>
<td><strong>Algorithm</strong></td>
<td>122</td>
<td>126</td>
<td>123</td>
</tr>
<tr>
<td><strong>Random</strong></td>
<td>125</td>
<td>121</td>
<td>107</td>
</tr>
</tbody>
</table>
## The Survey

### The DJ

<table>
<thead>
<tr>
<th>Playlist guess</th>
<th>Guess</th>
<th>Actual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human Expert</td>
<td>3.6</td>
<td>3.5</td>
</tr>
<tr>
<td>Algorithm</td>
<td>3.0</td>
<td>2.25</td>
</tr>
<tr>
<td>Random</td>
<td>2.0</td>
<td>3.25</td>
</tr>
</tbody>
</table>
Objective Evaluation
Some playlist stats

### Playlist stats

<table>
<thead>
<tr>
<th>Source</th>
<th>Radio Paradise</th>
<th>Musicmobs</th>
<th>art of the mix</th>
<th>Pandora</th>
</tr>
</thead>
<tbody>
<tr>
<td>Playlists</td>
<td>45,283</td>
<td>1,736</td>
<td>29,164</td>
<td>94</td>
</tr>
<tr>
<td>Unique Artists</td>
<td>1,971</td>
<td>19,113</td>
<td>48,169</td>
<td>556</td>
</tr>
<tr>
<td>Unique Tracks</td>
<td>6,325</td>
<td>93,931</td>
<td>218,261</td>
<td>908</td>
</tr>
<tr>
<td>Average Length</td>
<td>4.3</td>
<td>100</td>
<td>20</td>
<td>11</td>
</tr>
<tr>
<td>% with duplicate artist</td>
<td>0.3%</td>
<td>79%</td>
<td>49%</td>
<td>48%</td>
</tr>
<tr>
<td>% with consecutive artists</td>
<td>0.3%</td>
<td>60%</td>
<td>20%</td>
<td>5%</td>
</tr>
</tbody>
</table>

Pandora playlist stats based on listening on 44 separate ‘stations’
### Objective evaluation
Tag diversity

**Playlist Tag Diversity**

<table>
<thead>
<tr>
<th>Source</th>
<th>Tag Diversity</th>
<th>Random</th>
</tr>
</thead>
<tbody>
<tr>
<td>MusicMobs</td>
<td>0.29 / 0.18</td>
<td>0.51 / 0.13</td>
</tr>
<tr>
<td>Pandora</td>
<td>0.44 / 0.20</td>
<td>0.64 / 0.19</td>
</tr>
<tr>
<td>Art of the mix</td>
<td>0.48 / 0.17</td>
<td>0.61 / 0.11</td>
</tr>
<tr>
<td>Radio Paradise</td>
<td>0.75 / 0.13</td>
<td>0.75 / 0.13</td>
</tr>
</tbody>
</table>

Tag Diversity: unique artist tags vs. total artist tags
# Radio Paradise diversity examples

<table>
<thead>
<tr>
<th>Artist</th>
<th>Track</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Volt</td>
<td>Live Free</td>
<td>Alt-country, americana, rock, country, folk, indie</td>
</tr>
<tr>
<td>Sun Kil Moon</td>
<td>Gentle Moon</td>
<td>indie, folk, singer-songwriter, americana, Alt-country, alternative</td>
</tr>
<tr>
<td>ANi DiFranco</td>
<td>Angry Any More</td>
<td>folk, singer-songwriter, female vocalists, indie, alternative, rock</td>
</tr>
<tr>
<td>Jim White</td>
<td>Handcuffed to a fence in Mississippi</td>
<td>Alt-country, singer-songwriter, americana, folk, indie, country</td>
</tr>
<tr>
<td>Jess Klein</td>
<td>Soda Water</td>
<td>folk, female vocalists, singer-songwriter, indie, acoustic, girls with guitars</td>
</tr>
</tbody>
</table>

Diversity: 0.367
11 unique tags out of 30
Radio Paradise diversity examples

<table>
<thead>
<tr>
<th>Artist</th>
<th>Track</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big Head Todd &amp; The Monsters</td>
<td>It’s Alright</td>
<td>rock, alternative, jam band, prog rock, Jam, 90s</td>
</tr>
<tr>
<td>Joni Mitchell</td>
<td>Be Cool</td>
<td>folk, singer-songwriter, female vocalists, Canadian, classic rock, acoustic</td>
</tr>
<tr>
<td>Chet Baker</td>
<td>Tangerine</td>
<td>jazz, trumpet, cool jazz, blues, jazz vocals, easy listening</td>
</tr>
</tbody>
</table>

Diversity: 1.0
18 unique tags out of 18
# Pandora diversity examples

## Low Diversity Playlists

<table>
<thead>
<tr>
<th>Artist</th>
<th>Track</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Pitchfork</td>
<td>Timekiller</td>
<td>industrial, ebm, electronic, darkwave, Gothic, synthpop,</td>
</tr>
<tr>
<td>Covenant</td>
<td>We stand alone</td>
<td>melodic black metal, black metal, synthpop, metal, industrial, futurepop</td>
</tr>
<tr>
<td>Icon of Coil</td>
<td>Faith? Not Important</td>
<td>ebm, industrial, futurepop, electronic, synthpop, darkwave</td>
</tr>
<tr>
<td>Neuroticfish</td>
<td>Waving Hands</td>
<td>ebm, futurepop, industrial, synthpop, electronic, goth</td>
</tr>
<tr>
<td>Project Pitchfork</td>
<td>Momentum</td>
<td>industrial, ebm, electronic, darkwave, Gothic, synthpop</td>
</tr>
<tr>
<td>Covenant</td>
<td>Stalker</td>
<td>melodic black metal, black metal, synthpop, metal, industrial, futurepop</td>
</tr>
</tbody>
</table>

**Diversity: 0.305**

11 unique tags out of 36
## Pandora diversity examples

### High Diversity Playlists

<table>
<thead>
<tr>
<th>Artist</th>
<th>Track</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metallica</td>
<td>The Call of Ktulu</td>
<td><code>metal, thrash metal, heavy metal, rock, hard rock, metallica</code></td>
</tr>
<tr>
<td>Linkin Park</td>
<td>Pushing Me Away</td>
<td><code>rock, Nu Metal, alternative, metal, Linkin Park, punk</code></td>
</tr>
<tr>
<td>Creed</td>
<td>One Last Breath</td>
<td><code>rock, alternative, hard rock, Grunge, metal, punk</code></td>
</tr>
</tbody>
</table>

**Diversity:** 0.611

11 unique tags out of 18

Evanescence Radio
# Musicmobs diversity examples

## Low Diversity Playlists

<table>
<thead>
<tr>
<th>Artist</th>
<th>Track</th>
<th>Tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect Circle</td>
<td>(54 Tracks)</td>
<td>rock, alternative, Progressive rock, metal, hard rock, industrial</td>
</tr>
<tr>
<td>Tool</td>
<td>(43 Tracks)</td>
<td>Progressive metal, Progressive rock, metal, rock, alternative, Progressive</td>
</tr>
</tbody>
</table>

Diversity: 0.014
8 unique tags out of 582
Playlist Cohesion Metric

- Goal - find level of cohesion in an ordered sequence such as a playlist

- How:
  - Represent the item space as a connected graph
  - Find the shortest weighted path that connects the ordered sequence
  - Average step length is the cohesion index
I especially like the "playlist cohesion" metric on slide 185(!) -- I will definitely refer to this next time I make a mix tape for a girl.
• Consider [A, E, U, X]
• Distance: [3, 7, 6] = 16
• Average Distance: 5.33
Playlist Cohesion Metric

- Consider [A, E, U, X]
- Distance: \([3, 7, 6] = 16\)
- Average Distance: 5.33
Playlist Cohesion Metric

- Consider [A, E, U, X]
- Distance: [3, 7, 6] = 16
- Average Distance: 5.33
Playlist Cohesion Metric

- Consider [A, E, U, X]
- Distance: [3, 7, 6] = 16
- Average Distance: 5.33
Playlist Cohesion Metric

- Consider [A, E, U, X]
- Distance: [3,7,6] = 16
- Average Distance: 5.33
Playlist Cohesion Metric

- Consider [A, E, U, X]
- Distance: [3, 7, 6] = 16
- Average Distance: 5.33
Playlist Cohesion Metric

- Consider [Z, L, H, X]
- Distance: [15, 10, 9] = 34
- Average Distance: 11.3
Playlist Cohesion Metric

- Consider [Z, L, H, X]
- Distance: [15, 10, 9] = 34
- Average Distance: 11.3
• Consider [Z, L, H, X]
• Distance: [15, 10, 9] = 34
• Average Distance: 11.3
Consider [Z, L, H, X]

• Distance: [15, 10, 9] = 34

• Average Distance: 11.3
Playlist Cohesion Metric

- Consider [Z, L, H, X]
- Distance: [15, 10, 9] = 34
- Average Distance: 11.3
Consider [Z,L,H,X]

Distance: [15, 10, 9] = 34

Average Distance: 11.3
Building the graph
MusicBrainz Artist Relations

- Nodes are artists
- Edges are relations, weighted by significance
- 132 Relationship types. some examples:

<table>
<thead>
<tr>
<th>Edge type</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is Person</td>
<td>1</td>
</tr>
<tr>
<td>Member of band</td>
<td>10</td>
</tr>
<tr>
<td>Married</td>
<td>20</td>
</tr>
<tr>
<td>Performed with</td>
<td>100</td>
</tr>
<tr>
<td>Composed</td>
<td>250</td>
</tr>
<tr>
<td>Remixed</td>
<td>500</td>
</tr>
<tr>
<td>Edited Liner Notes</td>
<td>1000</td>
</tr>
<tr>
<td>Source</td>
<td>Average inter-song Distance</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Radio Paradise</td>
<td>0.08 / 0.06</td>
</tr>
<tr>
<td>Pandora</td>
<td>0.11 / 0.12</td>
</tr>
<tr>
<td>MusicMobs</td>
<td>0.13 / 0.10</td>
</tr>
<tr>
<td>Art of the mix</td>
<td>0.14 / 0.10</td>
</tr>
<tr>
<td>Random (RP)</td>
<td>0.27 / 0.22</td>
</tr>
<tr>
<td>Random (graph)</td>
<td>0.39 / 0.45</td>
</tr>
<tr>
<td>Random (AotM)</td>
<td>0.56 / 0.19</td>
</tr>
</tbody>
</table>
Building the graph
Echo Nest Artist Similarity

- Nodes are artists
- Edges are similar artists, weighted by similarity
# Echo Nest Artist Similarity Graph

<table>
<thead>
<tr>
<th>Source</th>
<th>Average inter-song Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pandora</td>
<td>1.57 / 1.4</td>
</tr>
<tr>
<td>Radio Paradise</td>
<td>2.27 / 1.0</td>
</tr>
<tr>
<td>MusicMobs</td>
<td>2.71 / 1.7</td>
</tr>
<tr>
<td>Art of the mix</td>
<td>3.02 / 1.4</td>
</tr>
<tr>
<td>Random (RP)</td>
<td>4.02 / 1.2</td>
</tr>
<tr>
<td>Random (AotM)</td>
<td>7.00 / 1.1</td>
</tr>
<tr>
<td>Random (graph)</td>
<td>7.89 / 1.78</td>
</tr>
</tbody>
</table>
The future of playlisting
DOUBLE-SIDED ROMANCE

SHARING MUSIC IS TOO EASY THESE DAYS.

I ALWAYS FELT RIPPED OFF BECAUSE I WAS TOO YOUNG TO ENJOY ACTUAL MIX TAPES.

EVEN IF THE MUSIC SUCKED, YOU KNEW THE OTHER PERSON DID SOME ACTUAL WORK.

JUST FOR YOU.

YOU COULD WRITE YOUR LOVE-TO-BE A CUSTOM MUSIC RECOMMENDATION ALGORITHM.

BUT THAT WOULD TAKE LONGER THAN I INTEND TO DATE HIM!

(C) 2010 R STEVENS :: DIESELSWEETIES.COM
Hybrid Radio
The Social Radio

• Produce playlists via weighted distance paths
• Next destination song is determined via a vote across all listeners
• Candidate songs selected from disparate communities
Hybrid Radio

Ratings

• Ratings are applied to the edge that lead to the song
• Song ratings -> playlist ratings
• Serving 2 purposes
  • Direct evaluation of playlists
  • Object based filtering
Convergence

When the cloud provide all the music and ubiquitous internet provides it all the time recommendation and playlisting merge
Convergence

The celestial jukebox needs a DJ.
The anonymous programmers who write the algorithms that control the series of songs in these streaming services may end up having a huge effect on the way that people think of musical narrative—what follows what, and who sounds best with whom. Sometimes we will be the d.j.s, and sometimes the machines will be, and we may be surprised by which we prefer.