Simple, open music recommendations

Sam Thursfield, GUADEC 2021
Recommenders are everywhere

assigned military aged male at birth @AliceAvizandum · Jul 8, 2018

TWITTER RECOMMENDATION ALGORITHM: would you like to see some porn your friends like

FACEBOOK RECOMMENDATION ALGORITHM: this terrible thing happened a year ago

AMAZON RECOMMENDATION ALGORITHM: buy five more TVs

YOUTUBE RECOMMENDATION ALGORITHM: would you like to become a nazi

@AliceAvizandum, Twitter
Can I make a recommendation algorithm?
Recommendation basics:

data → process → more data
Recommendation basics:

Music collection
Social data

Listen history
Audio analysis

→ process →

List of songs

<table>
<thead>
<tr>
<th>Title</th>
<th>Artist</th>
<th>Album</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Writing At The Gate</td>
<td>Love Grorer</td>
<td>Fresh Produce</td>
<td>4:47</td>
</tr>
<tr>
<td>She's</td>
<td>The Rubber Duck Orchestra</td>
<td>Jack Tamber's Album</td>
<td>4:30</td>
</tr>
<tr>
<td>Shrimp</td>
<td>Mr. Scruff</td>
<td>Trouser Jazz</td>
<td>7:01</td>
</tr>
<tr>
<td>Monkey Boogie</td>
<td>Milancolin</td>
<td>For Monkeys</td>
<td>3:26</td>
</tr>
<tr>
<td>She Likes to Smile</td>
<td>Rokka Tene</td>
<td>In This Life</td>
<td>5:10</td>
</tr>
<tr>
<td>Random I Am</td>
<td>Milancolin</td>
<td>For Monkeys</td>
<td>4:40</td>
</tr>
<tr>
<td>Acid Tape Track</td>
<td>Squarepusher</td>
<td>Selection Sixten</td>
<td>3:53</td>
</tr>
<tr>
<td>Bing Bong</td>
<td>Super Furry Animals</td>
<td>PZ/Vol. 2</td>
<td>5:18</td>
</tr>
<tr>
<td>Balkani Gaze</td>
<td>The Rubber Duck Orchestra</td>
<td>Jack Tamber's Album</td>
<td>4:54</td>
</tr>
<tr>
<td>Politicians In My Eyes</td>
<td>Death</td>
<td>For The Whole World To See</td>
<td>5:52</td>
</tr>
<tr>
<td>Trans Universal Express</td>
<td>Tom Parise</td>
<td>PZ/Vol. 2</td>
<td>2:52</td>
</tr>
</tbody>
</table>
Music recommendation basics:

playlist → process → playlist
Music recommendation basics:

playlist → process → playlist

```bash
> cpe tracker tracks | cpe shuffle --count 5 - | jq '{ title: .title, creator: .creator }' -c
{
  "title": "Find the River", "creator": "R.E.M."
}
{
  "title": "Good Good Things", "creator": "Descendents"
}
{
  "title": "Yuko and Hiro", "creator": "Blur"
}
{
  "title": "The Basset Hound’s Lament", "creator": "Thomas Truax"
}
{
  "title": "Widowmaker", "creator": "The Impossibles"
}
```

cpe = Calliope – set of commandline tools that work with playlists
playlist = list of JSON objects
Playlist format

XSPF ("spiff")

XSPF Home

XSPF is the XML format for sharing playlists.

- It is portable. You should be able to carry a playlist from one place to another.
- It is well-engineered. Most playlist formats have bugs that make life harder for programmers and users.
- It is free (as in liberty) and open. No proprietary lock-in.

Resolve the playlist in order to listen.
Demo: resolve a playlist

cpe tracker resolve-content five-songs.cpe | jq

You can now export the playlist as .m3u, .xspf, etc.

requires a free Spotify API key.
You can now upload the playlist to Spotify.

cpe spotify resolve five-songs.cpe | jq
Spotify facts

1600+ engineers

9000+ data pipelines

50 million tracks (25% of which have zero listens)

> half a trillion events captured per day

Source: https://research.atspotify.com/
Data sources

MusicBrainz: metadata
ListenBrainz: listen history
Last.fm: tags, listen history
Spotify: acoustic analysis, playlists, listen history

...and many more...
Demo: listening history

```
cpe -v 3 lastfm-history --user ssam scrobbles | head -n 5
```

Last 5 tracks I listened to.

```
cpe lastfm-history artists --first-play-since='6 months ago' \ 
   --min-listens 10
```

Artists I discovered in the last 6 months.

See more examples at: https://calliope-music.readthedocs.io/en/latest/examples/
Demo: ‘forgotten songs’ playlist

cpe lastfm-history tracks --last-play-before='1 year ago' \  --min-listens=5 | \  cpe shuffle - --count 20

Now select 30 minutes worth of music:

cpe tracker resolve-content - | cpe select \  --constraint=type:playlist-duration,vmin:30m,vmax:30m - | \  cpe export --format=m3u -

See more examples at: https://calliope-music.readthedocs.io/en/latest/examples/
Music playlist generation by adapted simulated annealing

Steffen Pauws, Wim Verhaegh, Mark Vossen

Philips Research, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

Abstract

We present the design of an algorithm for use in an interactive music system that automatically generates music playlists that fit the music preferences of a user. To this end, we introduce a formal model, define the problem of automatic playlist gen-
Constraint-based local search

Table 4. Constraint set ‘typical’.

<table>
<thead>
<tr>
<th>Description</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>All different songs</td>
<td>pairs-global(1, n_{max}, 1, d(v) = { x \mid x \neq v })</td>
</tr>
<tr>
<td>Release in 1980-2001</td>
<td>each-global(1, n_{max}, 7, {1980, 2001})</td>
</tr>
<tr>
<td>≥ 20% Stevie Wonder</td>
<td>fraction-global(1, n_{max}, 3, {Stevie Wonder}, .2, 1)</td>
</tr>
<tr>
<td>≥ 10% Seal</td>
<td>fraction-global(1, n_{max}, 3, {Seal}, .1, 1)</td>
</tr>
<tr>
<td>≥ 10% Peter Gabriel</td>
<td>fraction-global(1, n_{max}, 3, {Peter Gabriel}, .1, 1)</td>
</tr>
<tr>
<td>≥ 10% Janet Jackson</td>
<td>fraction-global(1, n_{max}, 3, {Janet Jackson}, .1, 1)</td>
</tr>
<tr>
<td>≥ 10% Mariah Carey</td>
<td>fraction-global(1, n_{max}, 3, {Mariah Carey}, .1, 1)</td>
</tr>
<tr>
<td>≥ 20% Phil Collins</td>
<td>fraction-global(1, n_{max}, 3, {Phil Collins}, .2, 1)</td>
</tr>
<tr>
<td>≥ 40% R&amp;B</td>
<td>fraction-global(1, n_{max}, 5, {R&amp;B}, .4, 1)</td>
</tr>
<tr>
<td>≥ 40% Popular</td>
<td>fraction-global(1, n_{max}, 5, {Popular}, .4, 1)</td>
</tr>
<tr>
<td>2-3 different genres</td>
<td>cardinality-global(1, n_{max}, 5, 2, 3)</td>
</tr>
<tr>
<td>Different succ. genres</td>
<td>chain-global(1, n_{max}, 5, d(v) = { x \mid x \neq v })</td>
</tr>
<tr>
<td>Similar succ. tempi</td>
<td>chain-global(1, n_{max}, 8, d(v) = { x \mid \text{sim}(x, v) \in [0, 0.1] })</td>
</tr>
</tbody>
</table>

\texttt{cat tracks.cpe | cpe shuffle - | cpe select \}
\texttt{--constraint=type:playlist-duration,vmin:60m,vmax:120m \}
\texttt{--constraint=type:item-duration,vmin:0s,vmax:6m - > playlist.cpe}

\texttt{simpleai: https://pypi.org/project/simpleai/}
GNOME Music Ideas

Listenbrainz integration
Link to music download sites
Automatic Musicbrainz tagging
Show artist info and links
Suggest ‘artists you might like’
Generate recommendation playlists
Summary

• Recommenders are here to stay.
• **Calliope** lets you build lo-fi music recommendations... and more.

  https://calliope-music.readthedocs.io/

  https://gitlab.com/samthursfield/calliope/

  pip install calliope-music

• Same design can work for developing other recommenders: videos, web history, suggested apps, local files...
Thanks for watching!