State of the art of Music Recommender Systems and open challenges

March 12th, 2015
MTG - Universitat Pompeu Fabra, Barcelona

Eugenio Tacchini
(Università Cattolica di Piacenza, Mentor.FM)
Outline

• Introduction to Music Recommender Systems
• Common recommendation techniques
• Challenges and Trends
• Lesson learned with Mentor.FM
What is a recommender system?

- Recommender systems are personalized information agents that provide recommendations: suggestions for items likely to be of use to a user (Burke, 2007)

- An “item” is a general term used to indicate what a RS suggest to its users, it can be an object (e.g. a DVD or a book) but also a person (a Facebook friend to add, a Twitter user to follow, ...)

- Important research area since mid-1990s, both in industry and academia
Some examples in Industry
Is music recommendation a special problem?
MentorFM // the next soundtrack to your life

www.mentor.fm
• 2011-2012: a prototype for academic purposes, during my Ph.D.
• Nov. 2013: public beta in ~180 countries
• Music streaming partner: Deezer.com
• Just for WWW, iOS/Android app soon
Recommendation Techniques
Content-based filtering

- Recommendations are based on characteristics (content) of the items to recommend

- How it works:
  - Determine a set of features which describes items (e.g. the music genome project, see next slide)
  - Describe all the items (vectorial representation)
  - Create user profiles according to the items they liked in the past (rating system)
  - Suggest items similar to the ones liked in the past
Pandora and the music genome project

• Each song is represented by about 400 features; some examples:
  • *Electric guitar*
  • *Duo rapping*
  • *Disco influences*
  • *Female vocal*
  • *Latin Percussion*
  • *Sad Lyrics*
  • ...

• Each feature (gene) is weighted 0 to 5
Pandora and the music genome project

Example, vectorial representation of the song
**Beatles - Twist and Shout**

![Vector representation](image)
Collaborative filtering

- Recommendations for a user are based on the preferences of other users, no need for content analysis
- A rating system is needed, either explicit or implicit
  - explicit: e.g. ask users to rate artists/songs
  - implicit: infer preferences from behavior analysis, e.g. if user X listens to song A ten times a day, it means he likes it
Collaborative filtering

- Two main approaches:
  - User-based approach
    - look for similarities among users
  - Item-based approach
    - look for similarities among items
user-based approach
## Example

### Preferences Matrix

<table>
<thead>
<tr>
<th></th>
<th>The Beatles</th>
<th>The Chemical Brothers</th>
<th>Arcade Fire</th>
<th>The Killers</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>LIKE</td>
<td></td>
<td>LIKE</td>
<td>LIKE</td>
</tr>
<tr>
<td>Bob</td>
<td>LIKE</td>
<td>LIKE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alice</td>
<td></td>
<td>LIKE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tom</td>
<td></td>
<td></td>
<td>LIKE</td>
<td>LIKE</td>
</tr>
<tr>
<td>Anna</td>
<td>LIKE</td>
<td></td>
<td>LIKE</td>
<td></td>
</tr>
</tbody>
</table>
### Example

#### Playcount Matrix

<table>
<thead>
<tr>
<th>Users</th>
<th>The Beatles</th>
<th>The Chemical Brothers</th>
<th>Arcade Fire</th>
<th>The Killers</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>800</td>
<td>0</td>
<td>30</td>
<td>42</td>
</tr>
<tr>
<td>Bob</td>
<td>11</td>
<td>35</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Alice</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Tom</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>Anna</td>
<td>500</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>

Would Anna like "The Killers"?
Similarity computation, a simple approach

Playcount to boolean
if playcount > threshold then playcount = 1 (LIKE)
if playcount <= threshold then playcount = 0

threshold = 10 for top artists, threshold = 5 otherwise

Similarity computation: Jaccard index

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

2/4
### Example

**user similarities matrix**

<table>
<thead>
<tr>
<th></th>
<th>John</th>
<th>Bob</th>
<th>Alice</th>
<th>Tom</th>
<th>Anna</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>1</td>
<td>0.25</td>
<td>0</td>
<td>0.66</td>
<td>0.66</td>
</tr>
<tr>
<td>Bob</td>
<td>0.25</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>0.33</td>
</tr>
<tr>
<td>Alice</td>
<td>0</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tom</td>
<td>0.66</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.33</td>
</tr>
<tr>
<td>Anna</td>
<td>0.66</td>
<td>0.33</td>
<td>0</td>
<td>0.33</td>
<td>1</td>
</tr>
</tbody>
</table>
### Example

#### Playcount Matrix

<table>
<thead>
<tr>
<th></th>
<th>The Beatles</th>
<th>The Chemical Brothers</th>
<th>Arcade Fire</th>
<th>The Killers</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>800</td>
<td>0</td>
<td>30</td>
<td>42</td>
</tr>
<tr>
<td>Bob</td>
<td>11</td>
<td>35</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Alice</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Tom</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>Anna</td>
<td>500</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>

Would Anna like “The Killers”?

Anna’s neighbor
**Example**

*Playcount Matrix*

<table>
<thead>
<tr>
<th>users</th>
<th>The Beatles</th>
<th>The Chemical Brothers</th>
<th>Arcade Fire</th>
<th>The Killers</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>800</td>
<td>0</td>
<td>30</td>
<td>42</td>
</tr>
<tr>
<td>Bob</td>
<td>11</td>
<td>35</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Alice</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Tom</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>Anna</td>
<td>500</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>

*Probably yes! Because John likes them, let’s recommend them!*

Anna’s neighbor
Item-based approach
### Example

**Playcount Matrix**

<table>
<thead>
<tr>
<th>users</th>
<th>The Beatles</th>
<th>The Chemical Brothers</th>
<th>Arcade Fire</th>
<th>The Killers</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>800</td>
<td>0</td>
<td>30</td>
<td>42</td>
</tr>
<tr>
<td>Bob</td>
<td>11</td>
<td>35</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Alice</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Tom</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>Anna</td>
<td>500</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>
Similarity computation, a simple approach

Playcount to boolean
if playcount > threshold then playcount = 1 (LIKE)
if playcount <= threshold then playcount = 0

threshold = 10 for top artists, threshold = 5 otherwise

Similarity computation: Jaccard index

<table>
<thead>
<tr>
<th></th>
<th>Arcade Fire</th>
<th>The Killers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>threshold</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Jaccard index</td>
<td>2/3</td>
<td></td>
</tr>
</tbody>
</table>
### Example

*Artists’ similarities matrix*

<table>
<thead>
<tr>
<th></th>
<th>The Beatles</th>
<th>The Chemical Brothers</th>
<th>Arcade Fire</th>
<th>The Killers</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Beatles</td>
<td>1</td>
<td>0.25</td>
<td>0.5</td>
<td>0.25</td>
</tr>
<tr>
<td>The Chemical Brothers</td>
<td>0.25</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Arcade Fire</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td>0.66</td>
</tr>
<tr>
<td>The Killers</td>
<td>0.25</td>
<td>0</td>
<td>0.66</td>
<td>1</td>
</tr>
</tbody>
</table>
The TOP-N Recommendation problem
Example

*Which artists could we suggest to Anna?*

<table>
<thead>
<tr>
<th>users</th>
<th>The Beatles</th>
<th>The Chemical Brothers</th>
<th>Arcade Fire</th>
<th>The Killers</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>800</td>
<td>0</td>
<td>30</td>
<td>42</td>
</tr>
<tr>
<td>Bob</td>
<td>11</td>
<td>35</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Alice</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Tom</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>Anna</td>
<td>500</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>
Example

Which artists could we suggest to Anna?

<table>
<thead>
<tr>
<th>users</th>
<th>The Beatles</th>
<th>The Chemical Brothers</th>
<th>Arcade Fire</th>
<th>The Killers</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>800</td>
<td></td>
<td></td>
<td>42</td>
</tr>
<tr>
<td>Bob</td>
<td>11</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Alice</td>
<td>2</td>
<td>20</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Tom</td>
<td>5</td>
<td>2</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>Anna</td>
<td>500</td>
<td>0</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>

The killers, because they are similar to Arcade Fire!
Challenges
The devil is in the details
Really, the devil is in the details! :-}
Licensing issues
The “Cold Start” problem
Import/Infer Music Preferences from external sources
Some preference sources

- Facebook “Likes”
- Facebook Posts
- Twitter artists followed
- Twitter posts (tweets)
- Listening history (Last.FM, Deezer, .... )
Let’s compare three preference sources

<table>
<thead>
<tr>
<th></th>
<th>Facebook</th>
<th>Deezer</th>
<th>Last.FM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Like</td>
<td>24.21% **</td>
<td>20.00%</td>
<td>12.63%</td>
</tr>
<tr>
<td>Dislike</td>
<td>6.02%</td>
<td>4.32%</td>
<td>4.27% **</td>
</tr>
<tr>
<td>Skip</td>
<td>36.54%</td>
<td>26.72% **</td>
<td>30.40%</td>
</tr>
</tbody>
</table>

User’s Feedback on Mentor.FM
“What I do, not what I say”

(Dunning & Friedman, Practical Machine Learning)
Discussion

- Some hypotheses:
  - a FB “like” can represent a strong user-artist connection, but we should be aware of false positive errors, users could like artists also:
    - to build their social image
    - to help artists get popularity
    - for other, not music-related, activities
Discussion

• **False negative** errors affect, in general, CF algorithms but on Facebook they might have additional causes related to the “social image” issue, for example:
  
  • The artist isn’t cool enough (and I don’t want to share my real taste)
  
  • The artist suggests connections with a social group I don’t want to make public
Infer Music Preferences from other domains
Music Identity Portability
Your music identity according to Mentor.FM

Your indie-ness level is 75%

Senpai says: Almost close to Wes Anderson!

ELECTRONIC

ITALIAN

AMBIENT

DANCE

HIP-HOP

DOWNTempo

CHILLOUT

POP

TECHNO

RAP

IDM

EXPERIMENTAL

ELECTRO

TRiP-HOP

ALTERNATIVE

MINIMAL

ROCK

INDIE
<table>
<thead>
<tr>
<th></th>
<th>Rdio</th>
<th>Spotify</th>
<th>Deezer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Favourite artists</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Playlists</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Listening history</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Music Data Integration
Meg’s page on Deezer

(source: http://www.deezer.com/artist/71255)
Noemi’s page on Spotify

Italian
Noemi

French
Noemi

(source: https://play.spotify.com/artist/62C5P1caRIK12ndTkzNjIA)
Convert the FB ID of the French artist “Billie” into a Spotify ID using Echonest Rosetta Stone

**API Request**
api_key=.........&id=facebook:artist:
62098951319&format=json&bucket=id:spotify

**API Answer**
"response": {"status": {"version": "4.2", "code": 0, "message": "Success"}, "artist": {"foreign_ids": [{"catalog": "spotify", "foreign_id": "spotify:artist:7K1v3zQdCvnxHvelcbTcZ0"}], "id": "AR2G86V1187FB3EB2E", "name": "Billie"}}

7K1v3zQdCvnxHvelcbTcZ0 is the wrong Billie!
Explicit Vs. Implicit feedback
Explicit Ratings

- 1-5 ratings, with or without semantic explanation, e.g. rateyourmusic.com
- Binary ratings (like/dislike), e.g. YouTube
- Unary ratings (like), e.g. Facebook
Implicit Ratings

- Purchase data
- Consumption data (songs listened)
- Sharing data
- ...


When did the user express the preference?
Personal information
Vs.
Recommendation Accuracy
trade off
Overspecialization problem

suggestions are accurate, but too similar / obvious

if you like the Beatles, you might like.......John Lennon
Diversity
Novelty
Serendipity
Serendipity

“A propensity for making fortunate discoveries while looking for something unrelated” (Wikipedia)

Books should be randomly shelved to facilitate novel browsing (Grose & Line, 1968)

Looking in a haystack for a needle and discovering a farmers daughter (Comroe, 1976)

If you focus on your interests, then your interests are going to stay what they are (Toms, 2000)

Incidental information acquisition (Williamson, 1998)
Serendipity in Recommender Systems

Degree to which the recommendations are presenting items that are both attractive and surprising (Herlocker et al., 2005)
State of the art

Serendipity measures

• As the deviation form the result provided by a PPM (Murakami et al., 2008)

Serendipity and discovery in recommender systems

• Determine underexposition and propose (Abbassi, Z. et al. 2009)
• Propose “border” items (Onuma, K. et al. 2009)
• Mix features of previous liked items (Oku, K. & Hattori, F. 2011)
• The Auralist Framework (Cao Zhang, Y. et all., 2012)
• Unexpectedness based on the utility theory of economics (Adamopoulos P. & Tuzhilin, A., 2014)

Divulgative talks

• TED presentation about “filter bubbles”: http://www.ted.com/talks/eli_pariser_beware_online_filter_bubbles
Define clusters of music
Examples of Musical Worlds

nofx

punk rock punk
skapunk

Animal collective

beirut
broken social scene
andrew bird
tv on the radio
architecture in helsinki
bon iver
clap your hands say yeah

“What is a "musical world"?: an affinity propagation approach.”
(Tacchini, E., Damiani, E, 2011)
Which cluster might contain serendipitous music?
How to introduce the user to that new world?
Evaluation
Evaluation

- Some classic accuracy measures

  - MAE: \( \frac{\sum_{i=1}^{n} |P_i - R_i|}{n} \)
  
  - MSE: \( \frac{\sum_{i=1}^{n} (P_i - R_i)^2}{n} \)
  
  - RMSE: \( \sqrt{\frac{\sum_{i=1}^{n} (P_i - R_i)^2}{n}} \)

- Decision support evaluation

- A/B test
Trust / Reputation
Improve user-based with trust/reputation information

- Users having higher trust/reputation get additional weight
- One method to get trust/reputation data is via Social Network Analysis
CUTTING-EDGE CHALLENGES
Music + Talk
Explain unexpected connection
Can I recommender system suggest something REALLY new?
Thanks!
eugenio.tacchini@gmail.com
eugenio@mentor.fm