CS6120: Intelligent Media Systems

Dr. Derek Bridge
School of Computer Science & Information Technology
UCC

Everything is Special

- Every product has its own special characteristics
- These must be taken into account when developing recommender systems

Why Music is Special

- Huge item space
  - e.g. 18 million songs on iTunes
- Very low cost per item
  - user can just skip a poor recommendation
- Many item types
  - tracks, albums, artists, genres, covers, remixes, concerts, labels, playlists, radio stations, other listeners, etc.

Paul Lamere: http://musicmachinery.com/2011/10/23/what-is-so-special-about-music/

Why Music is Special

- Low consumption time
  - users may need a lot of recommendations
- Very high per-item reuse
  - should recommend some already-consumed items
- Highly passionate users
  - users take offence at certain poor recommendations
- Highly contextual usage
  - requires context-sensitive recommendations

Paul Lamere: http://musicmachinery.com/2011/10/23/what-is-so-special-about-music/

Why Music is Special

- Consumed in sequences
  - playlists, mixtapes, DJ mixes, albums
- Large personal collections
  - a bedrock for user profiles
- Highly social
  - we share it; it expresses identity

Paul Lamere: http://musicmachinery.com/2011/10/23/what-is-so-special-about-music/

Recommending a Song

- Collaborative recommenders
  - explicit ratings
  - implicit ratings
    - purchasing, downloading, playing, skipping, favouriting, blocking,…
    - collected by what LastFM calls “scrobbling”
  - a major problem is knowing exactly which song is being rated
    - due to multiple versions, misspellings, typos, etc.

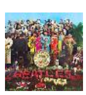
Recommending a Song

- Content-based recommenders
  - expert descriptions using a controlled vocabulary
    - expensive
  - end-user tagging
    - uneven
    - commonalities hidden by spelling errors, etc.
  - automatic audio analysis
    - extracts audio features such as tempo, rhythm, timbre, instrumentation,…

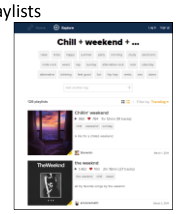
Recommending a Playlist

- Loss of 'expert' structure *("MP3 killed the radio star")*
  - the purchasing unit has changed: from album to single song
  - artistic effort (by bands, producers, DJs) to order tracks is being discarded



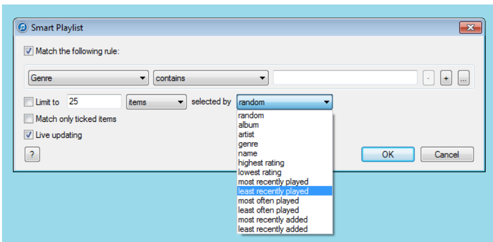


- But lots of end-user structure
  - E.g. individuals are creating and sharing playlists



Manual Playlist Creation in iTunes

Manual Playlist Creation: iTunes ‘smart’ playlists



Manual Playlist Creation

- Huge effort
  - dragging-and-dropping, or
  - defining rules
- Incomplete and vague tags and genres may result in low-quality Smart playlists
- Smart ‘playlists’ are sets of songs, not sequences of songs
- Having created them, how do you find the right one to play now?

Automatic Playlist Creation

- Audio analysis
  - E.g. B.Logan & A.Salomon:
    - user chooses a seed song
    - system generates a playlist using the songs most similar to this seed song
    - similarity is measured on the audio features
  - Critique: this playlist is a set, not a sequence
- User-based collaborative recommending
  - E.g. iTunes Genius
  - http://www.technologyreview.com/view/419198/how-itunes-genius-really-works/

Automatic Playlist Generation: Reusing Existing Playlists

- Users contribute playlists to LastFm, iTunes
  - other sources could be radio programs, web streams, music compilations, DJ sessions
  - a valuable resource
- Presumably, these capture knowledge about which songs 'sound well' in sequence
- We can reuse this knowledge to create new playlists
  - like market basket analysis
  - we look at Claudio Baccigalupo's work

The Goal

- Given a user's seed song s and desired length l, the goal is to find playlist p such that
  - p contains s
  - p is of length l
  - p is varied (does not repeat artist/album or, if it does, then the repetitions are not close)
  - p is coherently ordered

Reusing Existing Playlists

- He obtained a large collection of playlists from the web
- User-authored playlists are very often sets of songs, not sequences, so he excluded
  - very short lists
  - very long lists
  - alphabetically-ordered lists
  - …

Overview of his system

- Offline (in advance), analyse the playlists
  - find *patterns* (repeats of contiguous songs)
  - score them (e.g. by frequency)
- Online
  - ask user for seed song
  - retrieve playlists that contain that song
  - score them (e.g. based on the patterns that occur in them)
  - take the k with highest scores
  - combine these k playlists

Offline: Playlist Analysis

- Search through playlists for patterns
  - seek sequences of two or more songs that occur in the same order more than once
  - each pattern is given a pattern score
    - more frequently occurring patterns get a higher score
    - but shorter patterns are penalised
    - and patterns with highly popular songs are penalised
- High frequency sequences are evidence of coherent ordering

Offline: Playlist Analysis

- Here we have
  - one pattern (length 2) that occurs 3 times
  - one pattern (length 3) that occurs 2 times

U2 *Numb*	Coldplay *Yellow*	Roxette *Joyride*	The Beatles *Help!*	
U2 *Numb*	Coldplay *Yellow*	R.E.M. *Stand*	Radiohead *Creep*	
U2 *Lemon*	U2 *Numb*	Coldplay *Yellow*	Coldplay *In my place*	Coldplay *Trouble*
Oasis *Wonderwall*	Coldplay *Yellow*	Roxette *Joyride*	The Beatles *Help!*	Travis *Sing*

Online: Playlist Retrieval

- Obtain seed song s from user
- Consider playlists in the collection that contain s
  - each one of these is given a playlist score, which depends on
    - variety
      - variety of a playlist is initially 1 but the playlist is penalised for every artist that is repeated within n_{artist} songs and every album that is repeated within n_{album} songs, etc.
    - pattern score
      - sum up the pattern scores for every pattern that occurs in the playlist
  - retrieve the k playlists that have the highest playlist scores

Online: Playlist Retrieval

- Suppose the seed song is U2's Numb
  - how do you think these will score?

U2 *Numb* | Coldplay *Yellow* | Roxette *Joyride* | The Beatles *Help!*

U2 *Numb* | Coldplay *Yellow* | R.E.M. *Stand* | Radiohead *Creep*

U2 *Lemon* | U2 *Numb* | Coldplay *Yellow* | Coldplay *In my place* | Coldplay *Trouble*

Oasis *Wonderwall* | Coldplay *Yellow* | Roxette *Joyride* | The Beatles *Help!* | Travis *Sing*

Online: Combining the k Playlists

- We want to use the k playlists to produce a new playlist, p, of length l
- Here's how:
  - Initially p contains just s
  - Repeat until p is long enough:
    - For every song s' in the k playlists, create two candidate extensions of p: one in which s' is added to the start of p; and one on which it is added to the end of p
    - Compute the playlist score of each candidate extension
    - Choose the candidate with the highest score; this becomes p

Online: Combining the k Playlists

- Suppose the seed song is U2's Numb and $k = 3$
  - Retrieved:
    - U2 *Numb* | Coldplay *Yellow* | Roxette *Joyride* | The Beatles *Help!*
    - U2 *Numb* | Coldplay *Yellow* | R.E.M. *Stand* | Radiohead *Creep*
    - U2 *Lemon* | U2 *Numb* | Coldplay *Yellow* | Coldplay *In my place* | Coldplay *Trouble*
  - We start with this: U2 *Numb*
  - What are the candidate extensions, and how well do they score?

Some Results

- In some experiments, he used
  - 30,000 playlists
  - $k = 50$ (number of retrieved playlists)
  - $l = 10$
  - large values for n_{artist} and n_{album} to discourage repetition

Example Playlists
Seed: American Pie (Don McLean)

Playlist
(with penalties for popularity):

- We're An American Band (VV.AA.)
- Sweet Home Alabama (Lynyrd Skynyrd)
- More Than a Feeling (Boston)
- Bad Moon Rising (Creedence Clearwater Revival)
- American Pie (Don McLean)
- Mr. Blue Sky (Electric Light Orchestra)
- Switch (Will Smith)
- This Love (Maroon 5)
- Walkie Talkie Man (Steriogram)
- Walkin' On The Sun (Smash Mouth)

Playlist
(without penalties for popularity):

- Behind These Hazel Eyes (Kelly Clarkson)
- Beverly Hills (Weezer)
- I Just Wanna Live (Good Charlotte)
- American Idiot (Green Day)
- American Pie (Don McLean)
- Hotel California (The Eagles)
- Cocaine (Eric Clapton)
- Emerald Eyes (Fleetwood Mac)
- Carry On Wayward Son (Kansas)
- Sweet Home Alabama (Lynyrd Skynyrd)

Example Playlists
Seed: Soldier (Destiny's Child)

Playlist (with penalties for popularity):

- Let Me Love You (Mario)
- Hush (LL Cool J)
- Red Carpet (Pause, Flash) (R. Kelly)
- Hot 2 Nite (New Edition)
- Wonderful (Ja Rule)
- My Prerogative (Britney Spears)
- Two Step (Ciara)
- Soldier (Destiny's Child)
- Only U (Ashanti)
- Pass Out (Ludacris)

Playlist (without penalties for popularity):

- Disco Inferno (50 Cent)
- Mockingbird (Eminem)
- Obsession (Frankie J)
- I Just Wanna Live (Good Charlotte)
- Boulevard Of Broken Dreams (Green Day)
- Since U Been Gone (Kelly Clarkson)
- Two Step (Ciara)
- Soldier (Destiny's Child)
- Drop It Like It's Hot (Snoop Dogg)
- Get Back (Ludacris)

Reflections

- Not personalised
  - user's only input is seed song
  - no use of long-term profile of interests
  - no use of feedback

Context-Awareness

- Context
  - a dynamic set of factors describing the current state of the user
  - can change rapidly

- Mood
- Time
- Activity


- Weather
- Location
- Companions


Context-Aware Collaborative Recommenders

- Based on acquiring ratings in different contextual conditions

- As usual, ratings can be explicit or implicit

Rating Matrix Sparsity

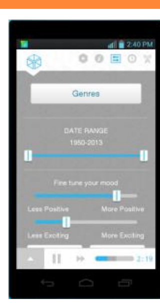
Snowing	Song 1	Song 2	Song3
Ann			

Raining	Song 1	Song 2	Song3
Ann		✓	

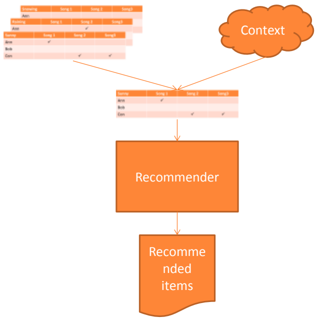
Sunny	Song 1	Song 2	Song3
Ann	✓		
Bob			
Con		✓	✓

Determining Context

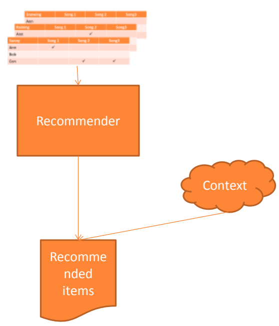
- Explicit
  - ask for it
- Implicit
  - sense it
- Inferred
  - work it out from other data



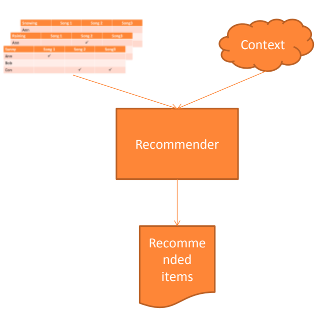
Context-Aware Recommenders: Contextual Pre-Filtering



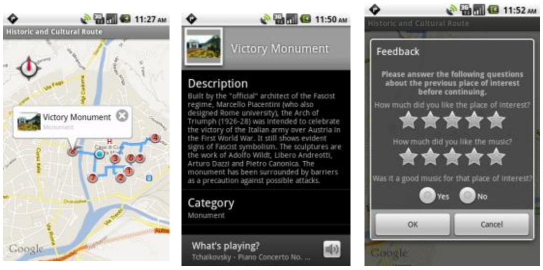
Context-Aware Recommenders: Contextual Post-Filtering



Context-Aware Recommenders: Contextual Modelling



Contextual Pre-Filtering Case Study: PlayingGuide Application



[M. Braunhofer et al., 2011. Recommending Music for Places of Interest in a Mobile Travel Guide]

The Approach

- Ways to identify music suited for a place-of-interest:
  - based on common *emotions* caused by listening to music and visiting places-of-interest
  - emotions represented as tags

Victory Monument

Heavy, Bright, Triumphant, Strong

Pyotr Tchaikovsky
Piano Concerto No. 1

The Approach

- Ways to identify music suited for a place-of-interest:
  - based on *semantic relations* between musicians and places-of-interest
  - relations mined from dbpedia

Vienna State Opera

Austrian opera composer; Was the director of Vienna State Opera

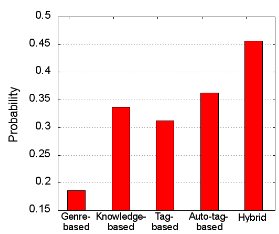
Gustav Mahler

User Study

- 58 users, 564 evaluation sessions

37

Results



- All context-aware approaches perform significantly better than the Genre-based recommendations
- The Hybrid approach produces the best results

38

Context-Aware Recommenders

- These recommenders are in their infancy
- But expect to see ever more of them!
  - e.g. article about Foursquare, http://readwrite.com/2014/03/17/foursquare-dennis-crowley-ceo-anticipatory-computing#awesm=~oyR4LUqPUqhDHa