Leveraging User's Mood and Song Popularity for Music Recommendation

Aritra Saha
Supervised by: Dr. Arnab Bhattacharya

Department of Computer Science & Engineering
IIT Kanpur

May 19, 2014

Outline

Outline

Motivation

User's Mood

- Determined by the features of the recently heard songs
- Usually constant over a short period of time and gradually changes

Song Popularity

- Trending songs as a useful recommendation
- Often overrides user's mood and preference

Motivation

User's Mood

- Determined by the features of the recently heard songs
- Usually constant over a short period of time and gradually changes

Song Popularity

- Trending songs as a useful recommendation
- Often overrides user's mood and preference

Contribution of this thesis

Mood Determination

- Optimize monitored rolling time duration
- Assigning weights to each contributing feature
- Use of Collaborative Filtering to determine similar users

Recommendation Tool

- Enables customization of weightages
- Recommends songs for a given user

Contribution of this thesis

Mood Determination

- Optimize monitored rolling time duration
- Assigning weights to each contributing feature
- Use of Collaborative Filtering to determine similar users

Recommendation Tool

- Enables customization of weightages
- Recommends songs for a given user

Outline

Related Work

Pandora Radio & Music Genome

- Categorzing each song by 300 - 500 genes on a scale of 0-5 with an increment of 0.5, for e.g., Pop/Rock, Jazz, Classical, level of guitar distortion, type of background vocals
- Manually analyzed by musicians with 20-30 minutes per song
- Vectors are matched by proprietary "matching algorithm"

Modelling Internet Radio - Yahoo!

- Internet Radio, a very close approximation of popular music
- Scraping data over time provides a variety of popularity statistics

Related Work

Pandora Radio & Music Genome

- Categorzing each song by 300 - 500 genes on a scale of 0-5 with an increment of 0.5, for e.g., Pop/Rock, Jazz, Classical, level of guitar distortion, type of background vocals
- Manually analyzed by musicians with 20-30 minutes per song
- Vectors are matched by proprietary "matching algorithm"

Modelling Internet Radio - Yahoo!

- Internet Radio, a very close approximation of popular music
- Scraping data over time provides a variety of popularity statistics

Data Structures & Algorithms

- Trie: A tree structure for prefix matching, used mostly for dictionary storage
- Levenshtein Distance: Measure distance between 2 strings with a score for addition, deletion & substitution
- Hungarian Algorithm: Optimal assignment for a given cost matrix
- Cosine Similarity: Given 2 vectors, a measure of similarity, ranges from -1 to 1

Data Acquisition

Million Song Dataset

- Compiled music data from multiple sources, for e.g., EchoNest, MusicBrainz, SecondHandSongs, Last.FM
- Has metadata as well as MFCC features with a disk occupancy of 250GB
- Cross-referenced tags from other music data websites for other information

Last.FM

- Fetch user's recent history, build up local database for collaborative filtering
- Get song's genre information to profile users' based on musical taste

Data Acquisition

Million Song Dataset

- Compiled music data from multiple sources, for e.g., EchoNest, MusicBrainz, SecondHandSongs, Last.FM
- Has metadata as well as MFCC features with a disk occupancy of 250GB
- Cross-referenced tags from other music data websites for other information

Last.FM

- Fetch user's recent history, build up local database for collaborative filtering
- Get song's genre information to profile users' based on musical taste

Outline

Loading Data

Load Million Song

- Million songs' commonly required information, for e.g., title & artist loaded up in a trie
- Song titles as the keys, enables faster searching while cleaning up user history data
- Further information of the songs are loaded into memory on demand to be used while recommending

User History

- 2,614 users' recent history has been compiled from Last.FM
- History has been cleaned up against the million songs

Loading Data

Load Million Song

- Million songs' commonly required information, for e.g., title & artist loaded up in a trie
- Song titles as the keys, enables faster searching while cleaning up user history data
- Further information of the songs are loaded into memory on demand to be used while recommending

User History

- 2,614 users' recent history has been compiled from Last.FM
- History has been cleaned up against the million songs

Profiling

Collaborative Filtering

- Based on the genres of the songs history, a normalized genre vector is populated which denotes the user's musical taste
- A set of 127 commonly found genres are considered for categorzing
- Cosine similarity mesaures the similarity between any 2 users
- Top k users' history are to be considered while recommending for a current user

Mood Determination

- Any m consecutive songs in a given history defines the mood at point in time
- A rolling window of m tracks is used to search for a similar mood in the top k similar users as that of the current user's present mood
- Given 2 mood windows, Hungarian algorithm determines the similarity score

Profiling

Collaborative Filtering

- Based on the genres of the songs history, a normalized genre vector is populated which denotes the user's musical taste
- A set of 127 commonly found genres are considered for categorzing
- Cosine similarity mesaures the similarity between any 2 users
- Top k users' history are to be considered while recommending for a current user

Mood Determination

- Any m consecutive songs in a given history defines the mood at point in time
- A rolling window of m tracks is used to search for a similar mood in the top k similar users as that of the current user's present mood
- Given 2 mood windows, Hungarian algorithm determines the similarity score

Song Features & Comparison

- Artist: 2 artists are compared in a similar way users are compared, by the use of consine similarity on their genre vectors
- Loudness: Logarithm of maximum power represented by the song. Ranges -100 to 100 dB
- Tempo: Average beats per minute. Determines the pace of the song. Ranges 0 to 500
- Popularity: Data provided as of 2010. Ranges 0 to 1

Work Flow

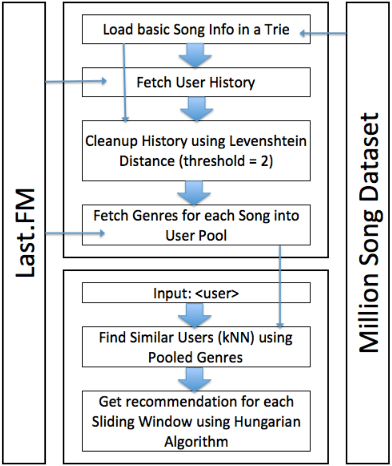

Figure 1: Work Flow

Outline

Weightages

Song Features

- Artist similarity has been assigned a weight of 60%
- 20% each for loudness & tempo
- These values have been evaluated to good results

Similarity v/s Popularity

- 65% weightage has been assgined to similarity and 35% to popularity
- A few number of test runs suggested the above weightages to be good

Weightages

Song Features

- Artist similarity has been assigned a weight of 60%
- 20% each for loudness & tempo
- These values have been evaluated to good results

Similarity v/s Popularity

- 65% weightage has been assgined to similarity and 35% to popularity
- A few number of test runs suggested the above weightages to be good

Performance Evaluation

- Most recent t tracks have been considered for testing
- Following m tracks are taken for current mood
- Recommended songs are then matched with the t tracks
- The rank of the top recommendation that appears in the test set is noted
- The similarity of the most similar mood window is also noted

Test Run 1

Similar Users	Mood Length	Weights	Confidence	Rank
50	5	$^1/_3, ^1/_3, ^1/_3$	62.89 %	944
75	5	$^1/_3, ^1/_3, ^1/_3$	48.45 %	3879
100	5	$^1/_3, ^1/_3, ^1/_3$	48.45 %	84
150	5	$^1/_3, ^1/_3, ^1/_3$	51.01 %	135
200	5	$^1/_3, ^1/_3, ^1/_3$	52.63 %	211
50	10	$^1/_3, ^1/_3, ^1/_3$	45.06 %	3418
75	10	$^1/_3, ^1/_3, ^1/_3$	45.50 %	4751
100	10	$^1/_3, ^1/_3, ^1/_3$	46.93 %	1722
50	5	$^1/_5, ^2/_5, ^2/_5$	43.28 %	4033
50	5	$^3/_5, ^1/_5, ^1/_5$	70.57 %	936
75	5	$^3/_5, ^1/_5, ^1/_5$	60.03 %	4367
100	5	$^3/_5, ^1/_5, ^1/_5$	62.10 %	78
150	5	$^3/_5, ^1/_5, ^1/_5$	64.73 %	120

Table 1: Test Results for Last.FM user: *3en*

Test Run 2

Similar Users	Mood Length	Weights	Confidence	Rank
50	5	$1/3, 1/3, 1/3$	62.39 %	2376
75	5	$1/3, 1/3, 1/3$	50.43 %	N/A
100	5	$1/3, 1/3, 1/3$	50.43 %	7608
150	5	$1/3, 1/3, 1/3$	50.43 %	8828
200	5	$1/3, 1/3, 1/3$	52.40 %	10018
50	10	$1/3, 1/3, 1/3$	48.91 %	N/A
75	10	$1/3, 1/3, 1/3$	48.91 %	N/A
100	10	$1/3, 1/3, 1/3$	48.91 %	N/A
50	5	$1/5, 2/5, 2/5$	43.28 %	N/A
50	5	$3/5, 1/5, 1/5$	70.70 %	2391
75	5	$3/5, 1/5, 1/5$	66.15 %	N/A
100	5	$3/5, 1/5, 1/5$	66.15 %	7095
150	5	$3/5, 1/5, 1/5$	66.15 %	7767

Table 2: Test Results for Last.FM user: *RJ*

Test Run 3

Similar Users	Mood Length	Weights	Confidence	Rank
50	5	$^1/_3, ^1/_3, ^1/_3$	62.59 %	59
75	5	$^1/_3, ^1/_3, ^1/_3$	50.43 %	607
100	5	$^1/_3, ^1/_3, ^1/_3$	48.37 %	736
150	5	$^1/_3, ^1/_3, ^1/_3$	51.94 %	1095
200	5	$^1/_3, ^1/_3, ^1/_3$	51.94 %	1428
50	10	$^1/_3, ^1/_3, ^1/_3$	48.91 %	2632
75	10	$^1/_3, ^1/_3, ^1/_3$	48.91 %	3736
100	10	$^1/_3, ^1/_3, ^1/_3$	46.91 %	4304
50	5	$^1/_5, ^2/_5, ^2/_5$	43.28 %	563
50	5	$^3/_5, ^1/_5, ^1/_5$	70.94 %	85
75	5	$^3/_5, ^1/_5, ^1/_5$	66.15 %	555
100	5	$^3/_5, ^1/_5, ^1/_5$	63.88 %	650
150	5	$^3/_5, ^1/_5, ^1/_5$	64.59 %	970

Table 3: Test Results for Last.FM user: *eartle*

Test Run 4

Similar Users	Mood Length	Weights	Confidence	Rank
50	5	$^1/_3, ^1/_3, ^1/_3$	61.84 %	141
75	5	$^1/_3, ^1/_3, ^1/_3$	49.83 %	629
100	5	$^1/_3, ^1/_3, ^1/_3$	49.71 %	674
150	5	$^1/_3, ^1/_3, ^1/_3$	51.10 %	4351
200	5	$^1/_3, ^1/_3, ^1/_3$	51.48 %	4363
50	10	$^1/_3, ^1/_3, ^1/_3$	47.49 %	3160
75	10	$^1/_3, ^1/_3, ^1/_3$	47.49 %	3135
100	10	$^1/_3, ^1/_3, ^1/_3$	47.54 %	3225
50	5	$^1/_5, ^2/_5, ^2/_5$	43.28 %	4422
50	5	$^3/_5, ^1/_5, ^1/_5$	67.74 %	103
75	5	$^3/_5, ^1/_5, ^1/_5$	66.18 %	470
100	5	$^3/_5, ^1/_5, ^1/_5$	64.43 %	471
150	5	$^3/_5, ^1/_5, ^1/_5$	64.43 %	5227

Table 4: Test Results for Last.FM user: *franhale*

Test Run 5

Similar Users	Mood Length	Weights	Confidence	Rank
50	5	$^1/_3, ^1/_3, ^1/_3$	62.64 %	4953
75	5	$^1/_3, ^1/_3, ^1/_3$	50.09 %	9857
100	5	$^1/_3, ^1/_3, ^1/_3$	50.09 %	11647
150	5	$^1/_3, ^1/_3, ^1/_3$	51.10 %	6587
200	5	$^1/_3, ^1/_3, ^1/_3$	51.48 %	8008
50	10	$^1/_3, ^1/_3, ^1/_3$	48.08 %	N/A
75	10	$^1/_3, ^1/_3, ^1/_3$	48.08 %	2584
100	10	$^1/_3, ^1/_3, ^1/_3$	48.08 %	2887
50	5	$^1/_5, ^2/_5, ^2/_5$	43.28 %	N/A
50	5	$^3/_5, ^1/_5, ^1/_5$	70.75 %	5005
75	5	$^3/_5, ^1/_5, ^1/_5$	65.97 %	7819
100	5	$^3/_5, ^1/_5, ^1/_5$	65.97 %	9345
150	5	$^3/_5, ^1/_5, ^1/_5$	68.19 %	5749

Table 5: Test Results for Last.FM user: *massdosage*

Optimizations

- Parallelization: Independent jobs have been forked in parallel to reduce runtime
- On-Demand Caching: Not only avoids loading the entire DB into memory, but also prevents disk access each time the same resource is called for. Also reduces multiple file accesses
- Minimal data handling: Minimal data is stored in memory in a serialized JSON format

Future Work

- Larger and newer dataset
- Machine learning to implement feedback mechanism for user specific weightages
- More features like MFCC can be included appropriately
- Code can be optimized even further by the use of distributed systems

Thank you!