Content-based Media Recomender Systems: Are we there yet?

Presentation by Stephen Travis Pope
stephen@FASTLabInc.com -- Oct 2008

Abstract

Measured in terms of the amount of time they’ve been heralded as the "next big thing," few technologies (hydrogen cars and cold fusion, perhaps) can rival content-based multimedia search engines. Using data features derived from multimedia content such as sound or images (without requiring human-generated metadata), together with advanced data-mining techniques to deliver user-preference-related similarity metrics (for search engines) has been a central topic in both image processing and music information retrieval for over a decade.

Abstract

The last year has seen the introduction (to great, and largely undeserved, fanfare) of a whole raft of music recommender systems. This presentation will introduce the topic of music recommender systems, and examine the feature extraction and data mining techniques that are at the core of all of these products. Concrete examples will be presented from the author’s own 4th-generation "SoLaTi" system, and 6th-generation SndsLike, and several products will be compared in terms of the play lists they recommend for given input songs.

Overview

- Introduction
- MMDB background
- Feature Extraction & Processing
- Segmenting and Seg-derived Features
- Dimensionality-reduction and Mapping
- Examples
  - SoLaTi (2007)
  - SndsLike (2012)

Music/Sound Database Projects

- ARA/DoubleTalk/HyperScore/MODE/Siren (1980–present)
  - Composer’s tools: metadata, persistency, data-mining
- Paleo (1996–9) MIDI performance expression data-mining
- NOLib (1998–9) Feature extraction framework in MATLAB
- BS Speech segmenter & database in Smalltalk (comps)
- MusicMagic, MusicIP, LibOFA, AmpliFind, GraceNote
- OMNI/LoCAA Network-based access, recommender (2001)
- FASTLab 2: Expert Mastering Assistant (EMA) (2002–4)
- FASTLab 3: Locus animation system (MUGI) (2006–7)
- FASTLab 4: SoLaTi recommender (Catalyst) (2007–8)
- FASTLab 5: Imagine Research/iZotope (SndObjRec) (2008–11)
- SndsLike & PlayListMgr (2012–3)
**Content Analysis for MMDBs**
- Feature vector & DB design
- First-pass analysis
  - Direct feature extraction
- Second-pass analysis (important!)
  - Smoothing, pruning, reduction
  - Higher-level features
- Numerical/statistical analysis (important!)
  - Avg/Dev, Histogram, GMM
- Machine-learning, data-mining
  - Clustering, classification, structure-learning

**Audio Feature Extraction**
- First-pass analysis (windowed)
  - Time-domain features
  - Frequency/chroma-domain features
- Second-pass analysis
  - Higher-level features, peak tracking
  - Perceptual mapping
  - Smoothing, pruning, reduction

**Audio Feature Extraction 1**
- First-pass (windowed) analysis
  - Time-domain features
    - Windowed RMS/peak amplitude (LF/HF bands)
    - Beats/tempo (AC, RR, model, tempo-changes)
    - H/LF RMS/Tempo AC & histogram stats
    - Silence detector
  - Frequency-domain analysis
    - Spectral coefficients & spectral measures
    - MFCC components
    - LPC coeff, noiselessness
    - Pitch-following (basis-esth)
    - Hi-Freq bands
    - Spectral-sub NoiseRed
  - Spatial/surround parameters
  - Populate rich/large 1st-stage feature vector

**Audio Feature Extraction 2**
- 2nd-pass analysis
  - Smoothing, pruning, reduction
  - Perceptual mapping
    - Loudness contour
    - Pitch, harmony and key
  - Higher-level features
    - Spectral peaks, tracks, SMS model
    - Spectral track statistics (rate of birth/death)
    - Tempo, tempo changes, tempo curve
  - Multi-pass: stage-configs and confidences

**Spectral Tracker Configurations**
- Each entry consists of a line with 4 data values:
  - peakWidth – closeness measure: peaks that close are considered to be one
  - minPeakAmplMeanClearanceRatio – the amplitude mean clearance ratio is defined as the ratio of a peaks amplitude to the mean amplitude of the peaks in the containing window. Only peaks with clearance ratios above this parameter are considered when finding tracks
  - birthFilterLevel – number of extra windows required to consider new peaks
    - “births.” A setting of 1 means it takes at least one more window with the peak ( 2 total ) to consider this peak as being born, etc.
  - deathFilterLevel – number of extra windows required to consider missing peaks
    - “deaths.” A setting of 1 means it takes at least one more window missing the peak ( 2 total ) to consider peak dead, etc.
- These settings were arrived at after much testing; others are possible
  - SpectralTrackerConfiguration { 1.06, 0.05, 0, 2 }
  - SpectralTrackerConfiguration { 1.04, 0.01, 0, 2 }
  - SpectralTrackerConfiguration { 1.02, 0.01, 0, 3 }
  - SpectralTrackerConfiguration { 1.01, 0.01, 0, 4 }

**Data Smoothing Examples**
- Bass pitch (sticky value island-builder)
- Tempo est. (multi-pass de-spiker, then GMM)
Audio Feature Extraction 3

- Feature vector statistics
  - Per-song feature average, mean, variance
  - Feature H-gram/GMM stats (val as main PDF ctr)
  - Feature vector pruning (strip meaningless data)

- Segmentation
  - Locate regularly spaced changes
  - Per-segment statistics, fade-in/out
  - Post-segmentation statistics

Music Segmentation

- Detect onsets
- Find regular hierarchy of onsets
- Segment track into verses
- Detect intro/outro
- Detect “solo” verse or bridge
- Calculate segmentation-related features
  (excellent genre/style correlation)

Audio Segmentation

- Basic (time-domain) procedure
  - Pick a feature vector weighting
  - Calculate inter-window distances (scalar)
  - Identify regular peak spacing

- Challenges
  - Tempo changes
  - Intro/outro
  - Click-track tempo
  - Compressed dynamic range
  - Finding the “1”
  - Aggressive (multi-weight, multi-tolerance blackboard) algorithm with confidence measure works ~85% of the time for our (very eclectic) test DB (1691 failed out of 14637) (allowing up to 30 segments)

Segmentation Techniques/Options

- Distance metrics and inter-segment-boundary detection
- Finding relevant segmentation
  - Grouping short segments
  - Dividing long segments
  - HMMS and Viterbi
  - Similarity regions
  - Simulated annealing
  - Blackboard systems

4 Song Segments

- Distance Weightings

- Average, dynamic range, spikiness
- Choose red or green (?)
The MIR song analysis process consists of a three-stage process: a learning/segmentation stage, a feature extraction stage that reads audio data and derives higher-level features (e.g., tempo or spectral tracks), followed by pruning data and derives higher-level features (e.g., typical/SoloIndex, TypicalStart, SoloStart, SoloCentroid, SoloVariety, SoloTempo, SoloDynRange from Isongs where title = 'I Believe In Love';

Summary

Improving Music Information Retrieval using Segmentations

Song Segmentation Data

<table>
<thead>
<tr>
<th>artist</th>
<th>title</th>
<th>segmentweight</th>
<th>numsegments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paula Cole</td>
<td>I Believe In Love</td>
<td>0.923772</td>
<td>0.24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>verselength</th>
<th>typicalstart</th>
<th>solostart</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.631119</td>
<td>0.280232</td>
<td>0.590672</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>s_centroid</th>
<th>s_variety</th>
<th>s_tempo</th>
<th>s_dynrange</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4991</td>
<td>0.001422</td>
<td>0.3360</td>
<td>0.654455</td>
</tr>
</tbody>
</table>

Song/Segmentation Features

Music Segmentation

- Automatic music segmentation using the "footsteps" algorithm, which identifies the boundaries of the verse, chorus, and other structural segments in a song.
- A content-based playlist-generation system was built which can use this kind of segmentation statistics and see if they look like fade-in/-out sections, or more like quiet sections.
- A content-based playlist-generation system by proposing new features.

Applications

- Configuration of a list of distance-weighting maps keyed by feature

Segmentation Confidence Measures

- How to compare segmentations
  - # of peaks per segment
  - # of segments per song (2-8)
  - % of song accounted for
  - % of peaks accounted for
  - Which weighting was used
  - Which tolerance was used
  - Weighted metric of these?

Advanced Segmentation

- Use derivative of distance vector?
- Adaptive feature weightings/tolerances
- Heuristic techniques
- Confidence calculus (multi-D)
- Robust tree-based segment percolation methods
- Post-segmentation statistics (can be quite valuable, when present)
FV Pruning/Storage

- How to handle invalid data
  - If song is silent (set x/y/z to NULL)
  - If tempo guess invalid (BH sums the same)
  - If MFCC/LPC data not reasonable
- If SegmentConfidence < s_threshold
- DB output: SQL or file-based
- Write 1-4 FV records to DB
  - Avg, var, solo, typical FV records
  - Write 1 FC record
- Top-level metadata, ptrs to FV data
- Normalize DB?

DB Processing Techniques

- Machine-Learning, data-mining, AI
  - Many techniques
  - Many apps
- Dimensionality reduction
  - PCA, ISA, SOM, SVM, trees, nets, ...
- Clustering, classification
  - Fixing incomplete/noisy classification
- Similarity metrics & matching

1st 2 PCA Dimensions

0.18 MFCCCoeff5 + 0.18 MFCCCoeff6 + 0.18 MFCCCoeff4 + 0.18 MFCCCoeff3 + 0.18 MFCCCoeff2 + 0.18 SpectralRolloff + 0.18 SpectralSlope + 0.18 SpectralFlux + 0.18 SpectralCentroid + 0.18 SpectralBandMax + 0.18 SpectralBandMin + 0.18 SpectralBandMid – 0.378 BSHUM3 – 0.345 LowPeakAmp – 0.323 BSHUM1 – 0.309 BSHUM2 – 0.298 HighPeakAmp – 0.294 fp_bass – 0.269 TempoAvg – 0.267 ZeroCrossings – 0.267 ZeroCrossingsVar – 0.237 HPRMS + 0.161 TempoWeight + 0.161 TempoAvg – 0.15 fp_gravity – 0.12 LRPRMSVar + 0.114 HighPeakBPM + 0.114 fp_focus + 0.093 LowPeakBPM + 0.086 HRPRMSVar – 0.058 SpectralFluxVar + 0.054 QualitySections + 0.053 LRPRMS – 0.049 SpectralBandMaxVar – 0.049 RMSVar – 0.045 PeakVar – 0.041 SoloTempo

Normalization Table

<table>
<thead>
<tr>
<th>Feature</th>
<th>Min</th>
<th>Max</th>
<th>Average</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoloTempo</td>
<td>0.049</td>
<td>0.179</td>
<td>0.045</td>
<td>0.041</td>
</tr>
<tr>
<td>RMSVar</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>PeakVar</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>SpectralVarietyVar</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>QuietSections</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>LowPeakBPM</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>HPRMS</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>LRPRMSVar</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>LRPRMS</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>RMSVar</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>PeakVar</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>SpectralFlux</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>SpectralCentroid</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>SpectralBandMax</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>SpectralBandMin</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>SpectralBandMid</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>BSHUM3</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>BSHUM1</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>BSHUM2</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>HighPeakBPM</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>HighPeakAmp</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>LowPeakAmp</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>fp_bass</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
<tr>
<td>fp_focus</td>
<td>0.179</td>
<td>0.345</td>
<td>0.179</td>
<td>0.179</td>
</tr>
</tbody>
</table>

Feature Rank (InfoGain)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Information Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoloCentroid</td>
<td>0.02184</td>
</tr>
<tr>
<td>SoloRMS</td>
<td>0.020293</td>
</tr>
<tr>
<td>SoloTempo</td>
<td>0.01918</td>
</tr>
<tr>
<td>SoloDynRange</td>
<td>0.01901</td>
</tr>
<tr>
<td>SoloStart</td>
<td>0.01772</td>
</tr>
<tr>
<td>SoloEnd</td>
<td>0.01640</td>
</tr>
<tr>
<td>SoloDuration</td>
<td>0.01568</td>
</tr>
<tr>
<td>SoloRange</td>
<td>0.01468</td>
</tr>
<tr>
<td>SoloPeak</td>
<td>0.01359</td>
</tr>
<tr>
<td>SoloLow</td>
<td>0.01248</td>
</tr>
<tr>
<td>SoloMax</td>
<td>0.01168</td>
</tr>
<tr>
<td>SoloMin</td>
<td>0.01076</td>
</tr>
<tr>
<td>SoloAvg</td>
<td>0.01001</td>
</tr>
<tr>
<td>SoloVar</td>
<td>0.00991</td>
</tr>
</tbody>
</table>

1st 2 ART-Tree-learning Features

- MFCCCoeffVar < 0.212848
- HPRMS < 0.08678
- LRPRMS < 0.08678
- SoloStart < 0.01531
- SoloEnd > 0.01531
- SoloDuration > 0.01531
- SoloRange > 0.01531
- SoloPeak > 0.01531
- SoloLow > 0.01531
- SoloMax > 0.01531
- SoloMin > 0.01531
- SoloAvg > 0.01531
- SoloVar > 0.01531
- NumSegments > 0
- SoloVariety < 0.00991
- SoloVariety > 0.00991
- Rock | 3.0/1.0
- Blues | 3.0/1.0
- Rock-Alternative | 3.0/1.0
- Jazz-Big Band-Swing | 3.0/1.0
- Comedy | 3.0/1.0
Tree-training
Example: Mulcher CART trees

Rule-learning

fp_focus > 0.519652
AND fp_bass <= 0.273051
AND LPRMS <= 0.858613
AND ZeroCrossings <= 0.197117
AND FadeOut > 0.1 AND FadeOut <= 0.7
AND SpectralVarietyVar <= 0.021011
AND MFCCoeff1Var <= 0.393386
AND SpectralBandMaxVar <= 0.003615
→ Classical (78.0/1.0)

Experimental Results: CURE

Cross-genre Distances
Mixed – 2 Blondie, 2 Cat Stevens, 2 Bill Cosby
PCA-semi-weighted EMD

Example “Confusion” Matrix
Application: Recommender Systems

- Search-by-user-preference and automatic play-list generation
- Content access, play-list generation
- Song ID, feature extraction
- Similarity search/sort
- Play-list sequencing (arch, cresc, tempo, energy)
- Multimedia-related tools
  - Human-supplied metadata
  - Automatic metadata only

Music Recommender Systems

- (selected, in approx. order of release)
- MusicIP MyDJ (FMAK0++)
- QMUL SoundBite
- MIT/EchoNest MusicBrain/API
- FMAK/SoLaTi
- iLike
- Apple/Gracenote MusicGenius
- MS Zune 3.0

SoLaTi System

- FASTLab, Inc + Catalyst
- Based on FMAK 4.2 analysis kernel
- Assume only audio-derived metadata
- To be augmented with other sources in Rev 2
- FV Statistics
  - Aggressive smoothing, histograms, GMM
  - Store mean and variance FVs for “typical” and “solo” verses (or mean/var for song)
- Multiple similarity metrics
  - Configurable/PCA FV weighting
  - Euclidean/Earth-mover’s/Mahalanobis distance

Example SoLaTi Play-list

Key: Joni Mitchell -- A Case Of You -- Folk

Example SoLaTi Play-list 4

Blondie -- Rapture

- Talking Heads -- Once In A Lifetime -- Rock/New Wave
- Bruce Springsteen -- The River -- Rock
- Ben Harper -- Homeless Child -- Rock/Alternative
- Al Green -- It's Going To Take More Than Love -- R&B/Soul Blues
- Bill Withers -- Ain't No Sunshine -- Soul
- The Fleshtones -- The Great Gatsby -- Rock
- Lara's Theme --clarinet montage -- Classical
- The Vanishing -- clarinet -- Classical
- Tom Waits -- Rain Dogs -- Rock
- Chita Rivera -- The Rhythm Of The Rain -- Latin
- The Smiths -- This Charming Man -- Indie
- The Clash -- London Calling -- Punk

SndslsLike

- Talking Heads -- Once In A Lifetime -- Rock/New Wave
- Bruce Springsteen -- The River -- Rock
- Ben Harper -- Homeless Child -- Rock/Alternative
- Al Green -- It's Going To Take More Than Love -- R&B/Soul Blues
- Bill Withers -- Ain't No Sunshine -- Soul
- The Fleshtones -- The Great Gatsby -- Rock
- Lara's Theme -- clarinet montage -- Classical
- The Vanishing -- clarinet -- Classical
- Tom Waits -- Rain Dogs -- Rock
- Chita Rivera -- The Rhythm Of The Rain -- Latin
- The Smiths -- This Charming Man -- Indie
- The Clash -- London Calling -- Punk
The “Latest Features”

- Standard time- and freq-domain features
- HPF/LPF versions
- Many freq bands
- Chroma, harmonicity, MFCCs & spectral measures
- Sp-slope, spread, bandwidth, variety, kurtosis, roll-off...
- Spectral tracking and track birth/death stats (useful)
- Fluctuation pattern features (E Pampalk)
- Beat histograms (G Tzanetakis)
- Statistical Spectrum Descriptors (Lidy & Rauber)
- Several tempo estimates (BH + stats)
- Several bass pitch estimates (+ stats) + tracking
- Several chord/key pitch estimates (+ stats)
- Musical segmentation and segment-related features

The Latest Statistics

- Lots of feature-dependent smoothing
- Data mode: noisy, bi-modal, clicky, etc.
- Take Gaussian Mixture Models (GMM) of all features
- Save gmm-avg, main-lobe width/weight, bi-modality...
- Also save dev, del2
The latest distance metrics

- Using noisy labels
- Dimensionality reduction vs clustering
  - PCA
  - SVMs
  - CURE
  - FLDA
- FLDA training and clusterer app
- Train on a couple dozen well-known genres

SndsLike Development Process

- Smalltalk prototype in Siren
- Analysis core in C++: RMS & FFT features
- Wrapper in Python
- Call-outs to Java (SSD) and Octave (FP) code
- Higher-level features
  - Rhythm, key, bass line, SSDs, etc.
- Simple tests
  - Feature extraction
  - DB populate batches

Data Sets

- FASTLab - 14 kSongs, very diverse, “high-quality,” well-encoded
- LikeZebra - 250 kSongs pop/rock
- MegaTrax - 160 kSongs + stems
- AudioNet - 200 kSongs + stems
- Others (not public)

Smalltalk Tools for SndsLike

- Test GUI button panel
- In Python & Qt
- Various data plots
- Several tools: gnuplot, XL, etc.
- MySQL tests
- Demo "Player" GUI
- C++/JUCE

Testing/Demo GUIs

- Test GUI button panel
- In Python & Qt
- Various data plots
- Several tools: gnuplot, XL, etc.
- MySQL tests
- Demo "Player" GUI
- C++/JUCE

OPs/Test GUI
SndsLike Player

<table>
<thead>
<tr>
<th>Track Title</th>
<th>Artist</th>
<th>Album</th>
<th>Genre</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>This Flight Tonight</td>
<td>Jon Mitchell</td>
<td>Blue</td>
<td>Folk</td>
<td>2:53</td>
</tr>
<tr>
<td>Kaa</td>
<td>Jon Mitchell</td>
<td>Blue</td>
<td>Folk</td>
<td>4:05</td>
</tr>
<tr>
<td>9 Case Of You</td>
<td>Jon Mitchell</td>
<td>Blue</td>
<td>Folk</td>
<td>4:23</td>
</tr>
<tr>
<td>The Last Time</td>
<td>Jon Mitchell</td>
<td>Blue</td>
<td>Folk</td>
<td>4:37</td>
</tr>
<tr>
<td>Court And Spark</td>
<td>Jon Mitchell</td>
<td>Court And Spark</td>
<td>Folk</td>
<td>3:46</td>
</tr>
<tr>
<td>Help Me</td>
<td>Jon Mitchell</td>
<td>Court And Spark</td>
<td>Folk</td>
<td>3:22</td>
</tr>
<tr>
<td>People Partis</td>
<td>Jon Mitchell</td>
<td>Court And Spark</td>
<td>Folk</td>
<td>2:35</td>
</tr>
<tr>
<td>Same Situation</td>
<td>Jon Mitchell</td>
<td>Court And Spark</td>
<td>Folk</td>
<td>2:57</td>
</tr>
<tr>
<td>Car On A Hill</td>
<td>Jon Mitchell</td>
<td>Court And Spark</td>
<td>Folk</td>
<td>3:02</td>
</tr>
<tr>
<td>Dear To You</td>
<td>Jon Mitchell</td>
<td>Court And Spark</td>
<td>Folk</td>
<td>3:09</td>
</tr>
<tr>
<td>Just Like This</td>
<td>Jon Mitchell</td>
<td>Court And Spark</td>
<td>Folk</td>
<td>4:25</td>
</tr>
<tr>
<td>Raised On Robbery</td>
<td>Jon Mitchell</td>
<td>Court And Spark</td>
<td>Folk</td>
<td>3:07</td>
</tr>
</tbody>
</table>

Code Tour in Spyder

Marketing

- Production music houses “still don’t get it”
  - “Customers aren’t asking for this…”
- Performance Problems
  - Many versions of the same track with different instrumentations - “stems”
  - Same track with/without vocals
  - Cover songs

Lessons Learned

I still want it!
- …so please make me one that gets accepted by the on-line services...

They (production music houses, record labels, Gracenote, Apple, Adobe, …) still don’t get it.

Thank You!

Q/A?

stephen@FASTLabInc.com