Designing a Content-Based **Music Search Engine**

Gert Lanckriet

Computer Audition Lab

UC San Diego

gert@ece.ucsd.edu

Work with Doug Turnbull, Luke Barrington and David Torres

CIW - Yahoo! Research Latinoamerica

Friday, December 21, 2007

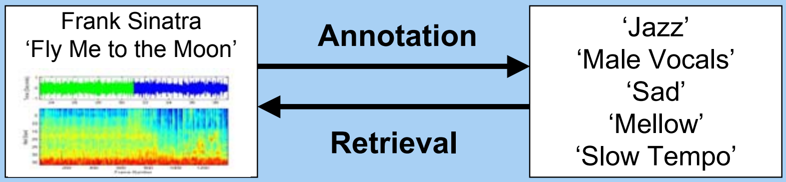
How do we find music?

- Query-by-Metadata - artist, song, album, year
  - We must know what we want
- Query-by-(Humming, Tapping, Beatboxing)
  - Requires talent
- Query-by-Song-Similarity
  - We must possess ‘acoustically’ similar songs
- **Query-by-Semantic-Description**
  - **Google** seems to work pretty well for **text**
  - **Semantic Image Labeling** is a hot topic in **Computer Vision**
  - Can it work for **music**?

Semantic Music Annotation and Retrieval

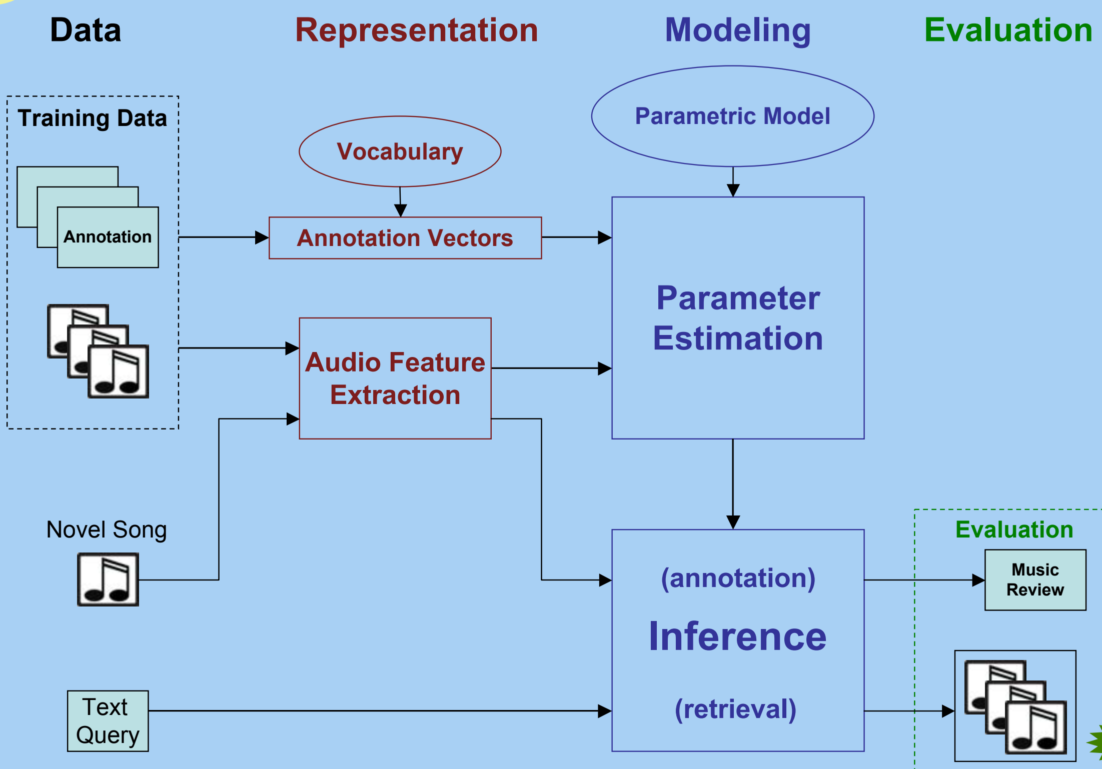
Our goal is build a system that can

1. **Annotate** a song with meaningful words
2. **Retrieve** songs given a text-based query



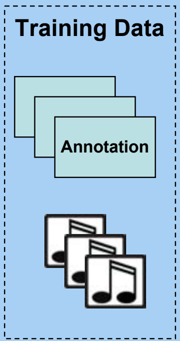
Plan: Learn a probabilistic model that captures a relationship between **audio content** and **words**.

System Overview



System Overview

Data

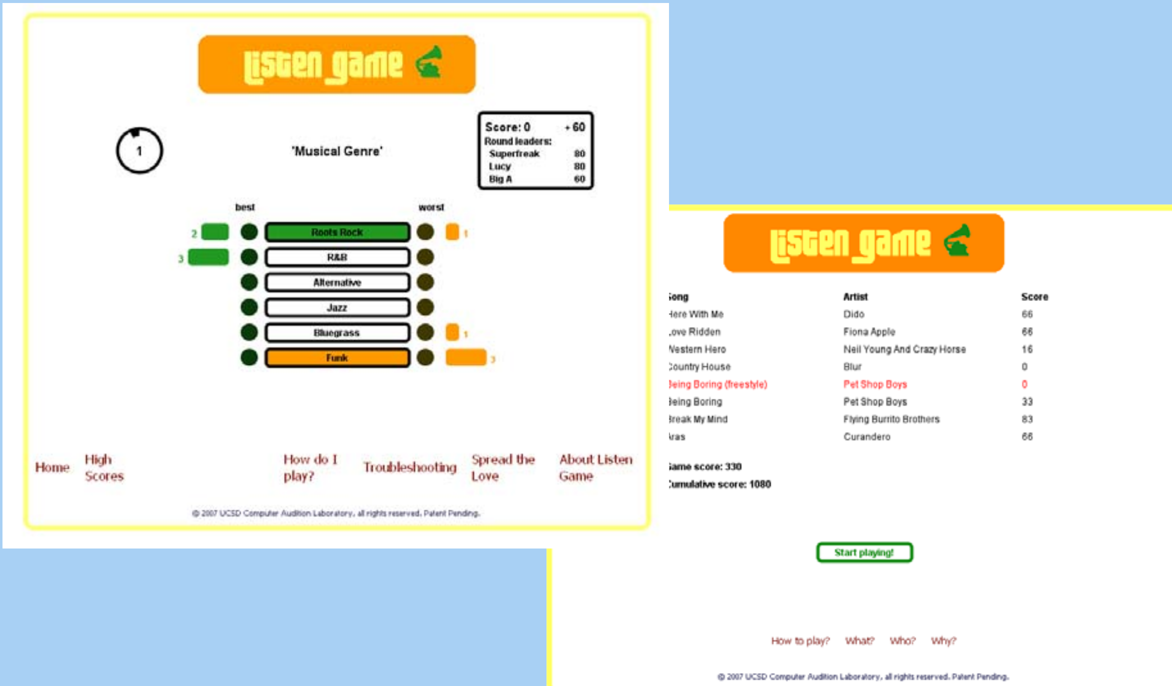


Collecting an Annotated Music Corpus

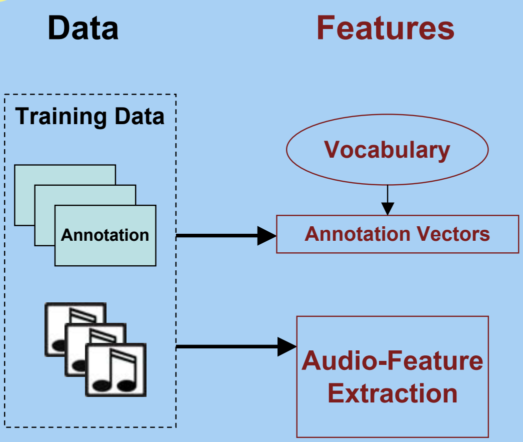
We have explored three techniques

1. Text-mining web documents
   - 2,100 song reviews from AMG All Music
   - Extracted a vocab of 317 words
2. Conducting a survey
   - 174-word hierarchical vocab - genre, emotion, usage, …
   - Paid 55 undergrads to annotate music for 120 hours
   - CAL500: 500 songs annotated by a minimum of 3 people
3. Deploying a ‘Human-Computation’ game
   - Web-based, multi-player game with real-time interaction
   - ESPGame by Luis Von Ahn
   - Listen Game

Listen Game Demo



System Overview



Semantic Representation: y

Choose vocabulary of 'musically relevant' words

- Instruments, Genre, Emotion, Rhythm, Energy, Vocal, Usages

Each annotation is converted to a real-valued vector

- 'Semantic association' between a word and the song.

Example: Frank Sinatra's "Fly Me to the Moon"

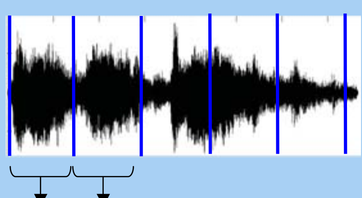
Vocab = {funk, jazz, guitar, female vocals, sad, passionate}

y = [0/4 , 3/4, 4/4 , 0/4 , 2/4, 1/4]

Acoustic Representation: X

Each song is represented as a bag-of-feature-vectors

- Pass a short time window over the audio signal
- Extract a feature vector for each short-time audio segment
- Ignore temporal relationships of time series



$$\mathbf{X} = \{ \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_t \}$$

Audio Features

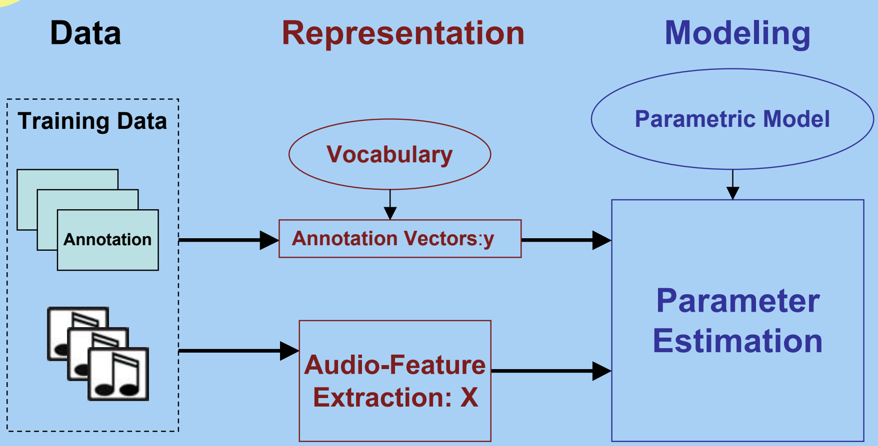
We calculate MFCC+Deltas feature vectors

- Mel-frequency Cepstral Coefficients (MFCC)
  - Low dimensional representation short-term spectrum
  - Popular for both representing speech, music, and sound effects
- Instantaneous derivatives (deltas) encode short-time temporal info
- 5,200 39-dimensional vectors per minute

Numerous other audio representations

- Spectral features, modulation spectra, chromagrams, …

System Overview



Statistical Model

Supervised Multi-class Labeling model

- Set of probability distributions over the audio feature space
- One Gaussian Mixture Model (GMM) per word - p(x|w)
- **Key Idea:** Estimate parameters for GMM using the set of training songs that are positively associated with the word

Notes:

- Developed for image annotation by Carneiro and Vasconcelos
- Scalable and Parallelizable
- Modified for real-value semantic weights rather than binary class labels
- Extended formulation to handle multi-word queries

Gaussian Mixture Model (GMM)

A GMM is used to model probability distributions over high dimensional spaces:

$$P(\mathbf{x}|w) = \sum_{r=1}^{R} \pi_r \mathcal{N}(\mathbf{x}|\mu_r, \Sigma_r)$$

A GMM is a weighted combo of R Gaussian distributions

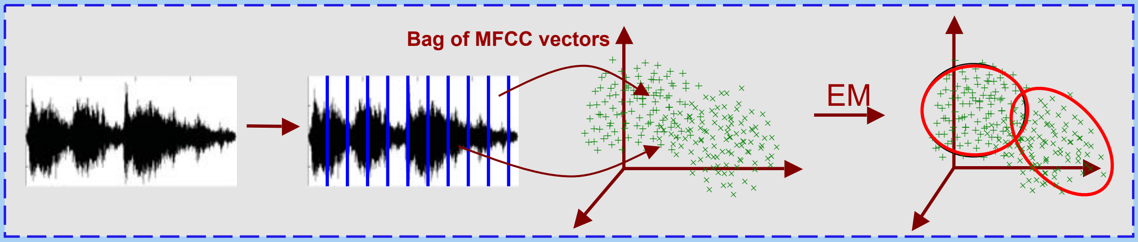
- π_r is the r-th mixing weight
- μ_r is the r-th mean
- Σ_r is the r-th covariance matrix

These parameters are usually estimated using a 'standard' **Expectation Maximization** (EM) algorithm.

Modeling Audio Content

Algorithm

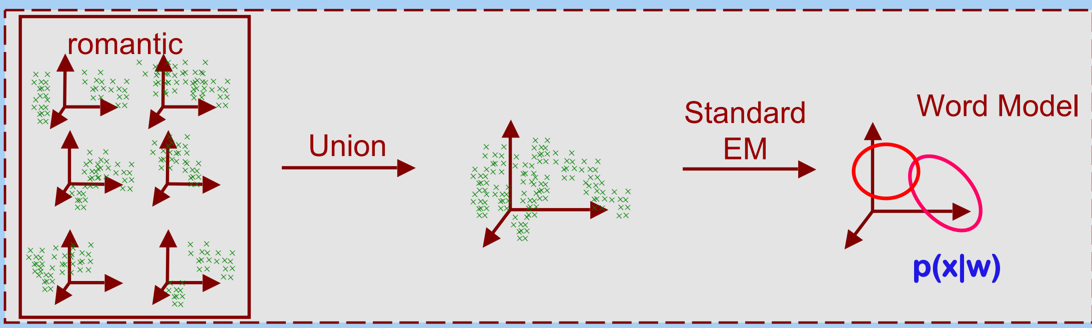
1. Segment audio signals
2. Extract short-time feature vectors
3. Estimate GMM using ‘standard’ EM



Three approaches for estimating p(x|w)

1. Direct Estimation

1. Identify songs associated with w
2. Union of feature vectors for these songs
3. Estimate GMM using ‘standard’ EM

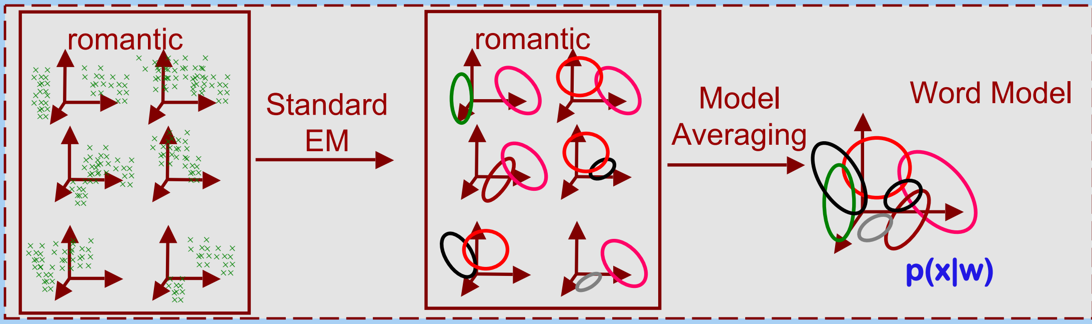


Problem: Direct Estimation is computationally difficult and empirically converges to poor local optima.

Three approaches for estimating p(x|w)

2. Model Averaging Estimation

1. Identify songs associated with w
2. Estimate a 'song GMM' for each song - p(x|s)
3. Use all mixture components from 'song GMMs'

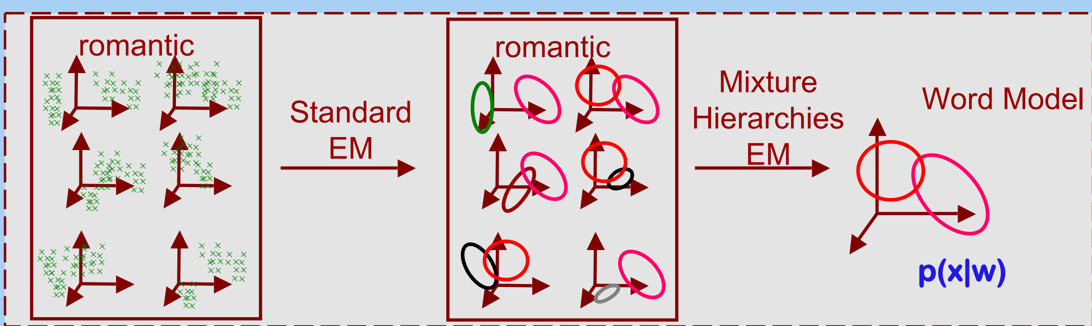


Problem: As the training set size grows, evaluating this distribution becomes prohibitively expensive.

Three approaches for estimating p(x|w)

3. Mixture Hierarchies

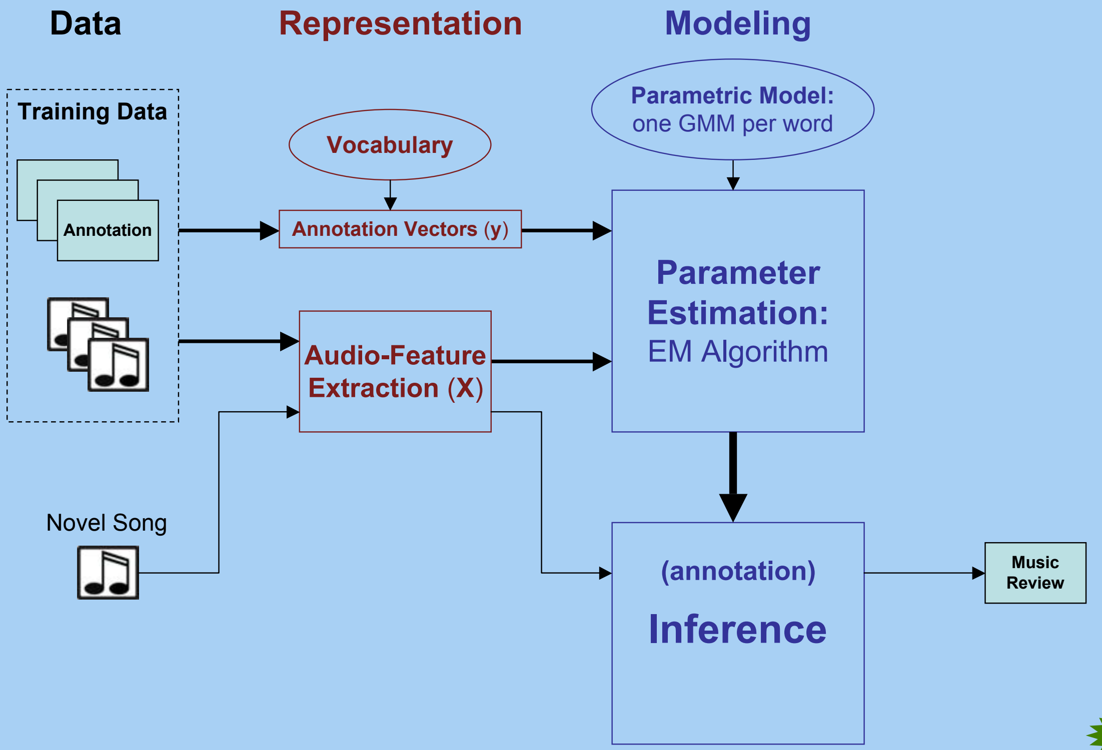
1. Identify songs associated with w
2. Estimate a 'song GMM' for each song - p(x|s)
3. Use the Mixture Hierarchies EM algorithm [Vasconcelos01]
   - Learn a 'mixture of mixture components'



Benefits

+ Computationally efficient for parameter estimation and inference
+ 'Smoothed' song representation → better density estimate

System Overview



Annotation

Given a novel song $X = \{x_1, \ldots, x_T\}$, calculate the probability of each word given the song:

$$P(w|X) = \frac{P(X|w)P(w)}{P(X)}$$

Assuming

1. Uniform word prior P(w)
2. Vectors are conditionally independent given a word

$$P(w|X) = \frac{\prod_{t=1}^{T} P(\mathbf{x}_t|w)}{\sum_{v \in V} \prod_{t=1}^{T} P(\mathbf{x}_t|v)}$$

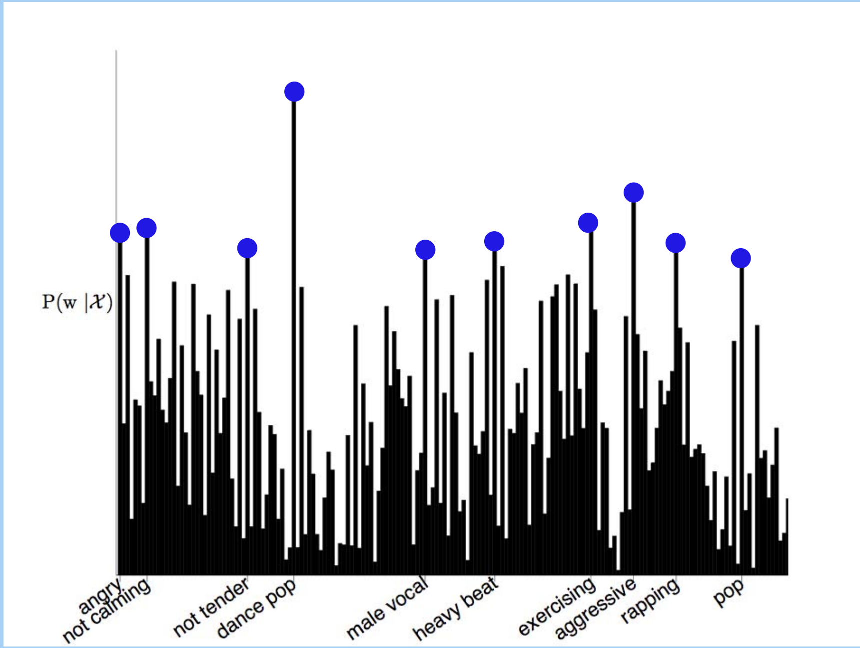
Semantic Multinomial:

- Conditional probabilities, P(w|X), defines multinomial over the vocabulary

Annotation: pick peaks of the semantic multinomial

Annotation

Semantic Multinomial for “Give it Away” by the Red Hot Chili Peppers



Annotation: Automatic Music Reviews

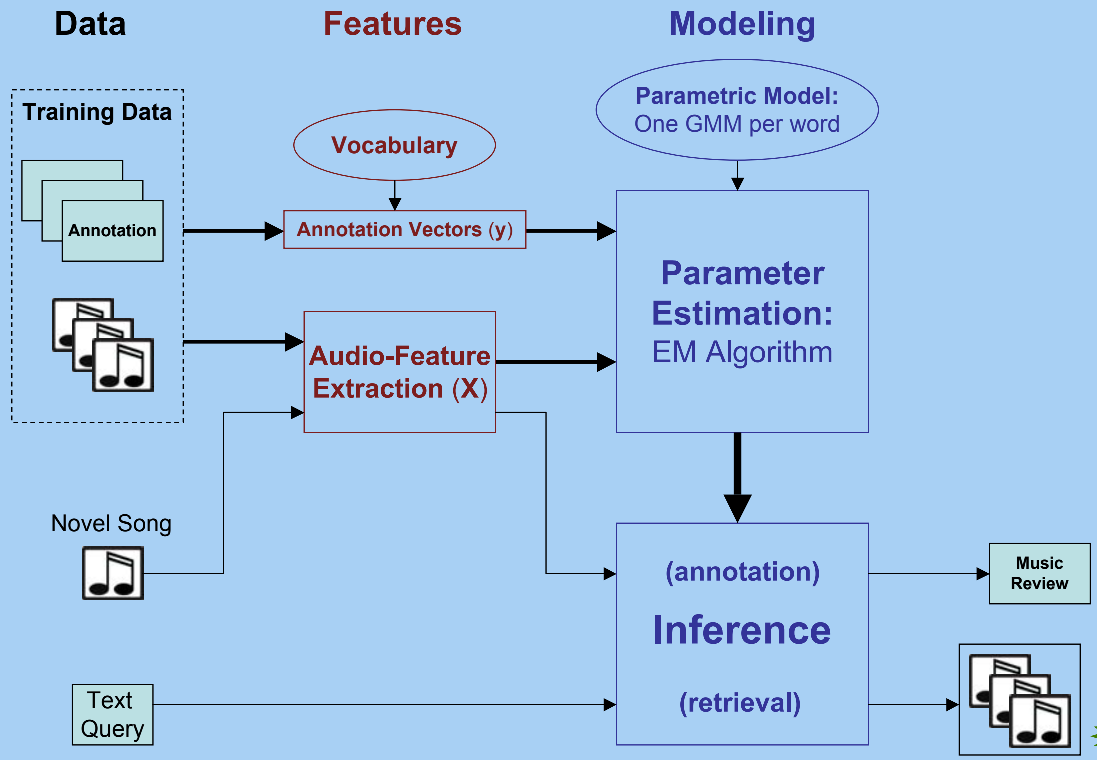
Dr. Dre (feat. Snoop Dogg) - Nuthin' but a 'G' thang

This is a **dance pop**py, **hip-hop** song that is **arousing** and **exciting**. It features **drum machine**, **backing vocals**, **male vocal**, a nice **acoustic guitar solo**, and **rapping**, **strong vocals**. It is a song that is very **danceable** and with a **heavy beat** that you might like listen to while **at a party**.

Frank Sinatra - Fly me to the moon

This is a **jazz**y, **singer / songwriter** song that is **calming** and **sad**. It features **acoustic guitar**, **piano**, **saxophone**, a nice **male vocal solo**, and **emotional**, **high-pitched** vocals. It is a song with a **light beat** and a **slow tempo** that you might like listen to while **hanging with friends**.

System Overview



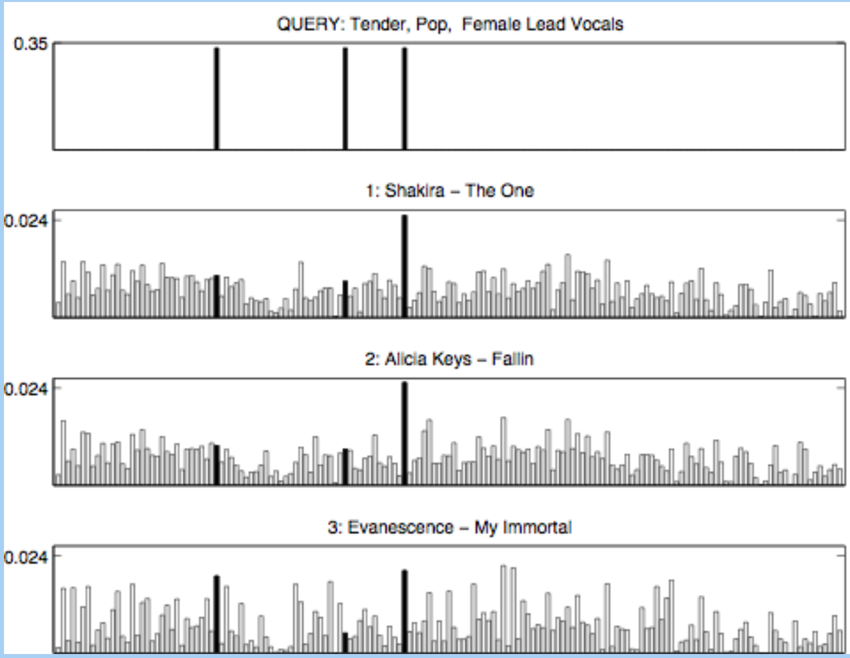
Retrieval

1. Annotate each song in corpus with a **semantic multinomial p**
   - p = {$P(w_1|X)$, …, $P(w_{|V|}|X)$}
2. Given a text-based query, construct a **query multinomial q**
   - $q_w = 1/|w|$, if word w appears in the query string
   - $q_w = 0$, otherwise
3. Rank all songs by the **Kullback-Leibler (KL) divergence**

$$KL(q||p) = \sum_{w \in V} q_w \log \frac{q_w}{p_w}$$

Retrieval

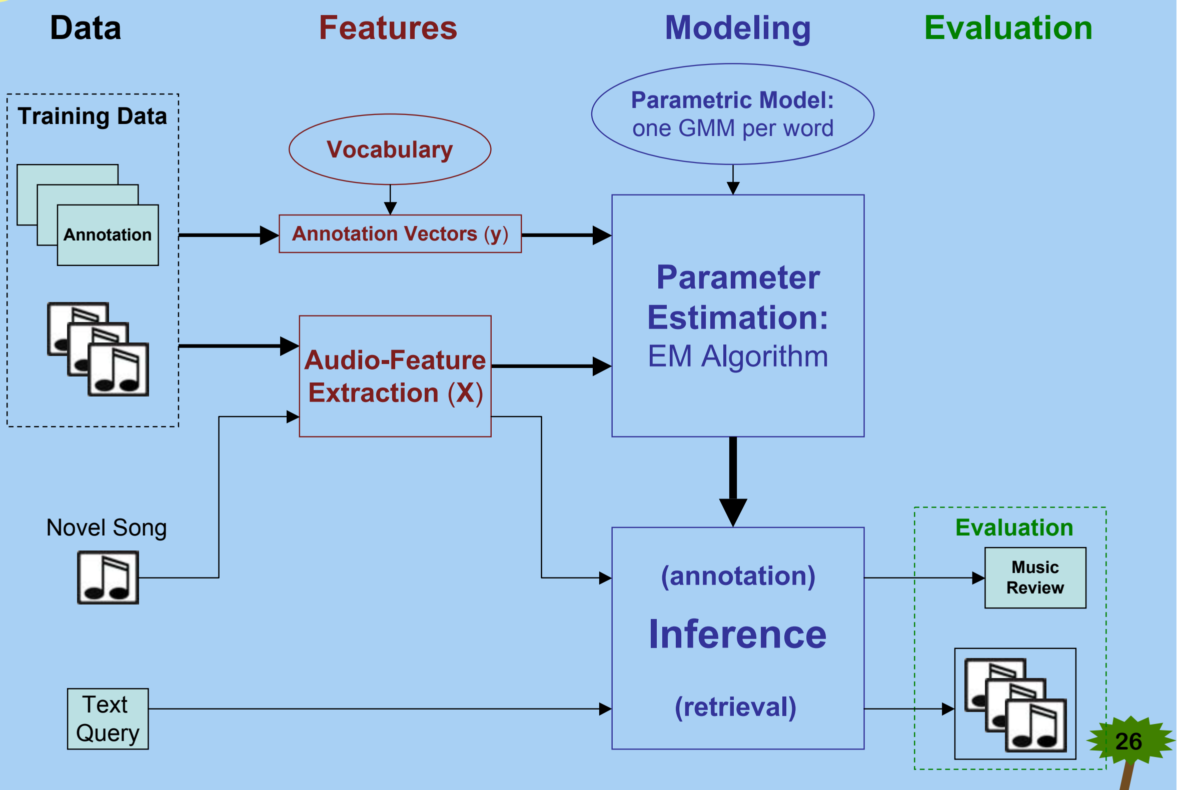
The top 3 semantic multinomials for the query "'pop', 'female lead vocals', 'tender'"



Retrieval: Query-by-Semantic-Description

Query	Retrieved Songs
'Tender'	**Crosby, Stills and Nash** - Guinevere **Jewel** - Enter from the East **Art Tatum** - Willow Weep for Me **John Lennon** - Imagine **Tom Waits** - Time
'Female Vocals'	**Alicia Keys** - Fallin' **Shakira** - The One **Christina Aguilera** - Genie in a Bottle **Junior Murvin** - Police and Thieves **Britney Spears** - I'm a Slave 4 U
'Tender' AND 'Female Vocals'	**Jewel** - Enter from the East **Evanescence** - My Immortal **Cowboy Junkies** - Postcard Blues **Everly Brothers** - Take a Message to Mary **Sheryl Crow** - I Shall Believe

System Overview



Quantifying Annotation

Our system annotates the Cal-500 songs with 10 words from our vocabulary of 174 words.

– 'Consensus Annotation' Ground Truth

Metric: 'Word' Precision & Recall

$$\text{Precision} = \frac{\text{\# songs correctly annotated with } w}{\text{\# songs annotated with } w}$$

$$\text{Recall} = \frac{\text{\# songs correctly annotated with } w}{\text{\# songs that should have been annotated } w}$$

Mean Word Recall and Word Precision are the averages over all words in our vocabulary.

Quantifying Annotation

Our system annotates the Cal-500 songs with 10 words from our vocabulary of 174 words.

Method	Precision	Recall
Random	0.14	0.06
Upper Bound	0.71	0.38
Our System	0.27	0.16
Human	0.30	0.15

Compared with a human, our model is

- worse on objective categories - instrumentation, genre
- about the same on subjective categories - emotion, usage

Quantifying Retrieval

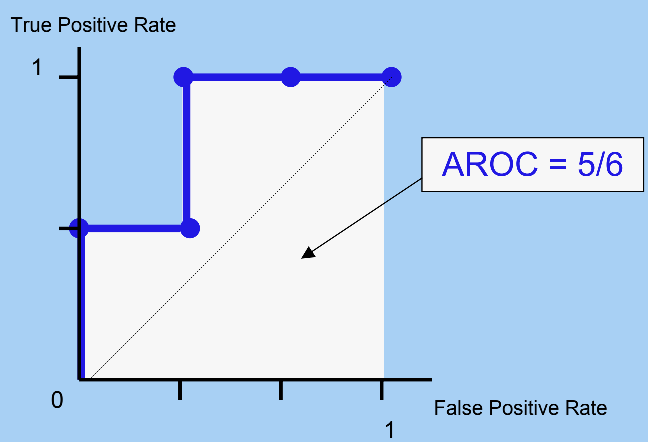
Rank order test set songs

– KL between a query multinomial and semantic multinomials
– 1-, 2-, 3-word queries with 5 or more examples

Metric: Area under the ROC Curve (AROC)

Rank by 'Romantic'

Rank	Label	TP	FP
1	R	1/2	0
2	-	1/2	1/3
3	R	1	1/3
4	-	1	2/3
5	-	1	1



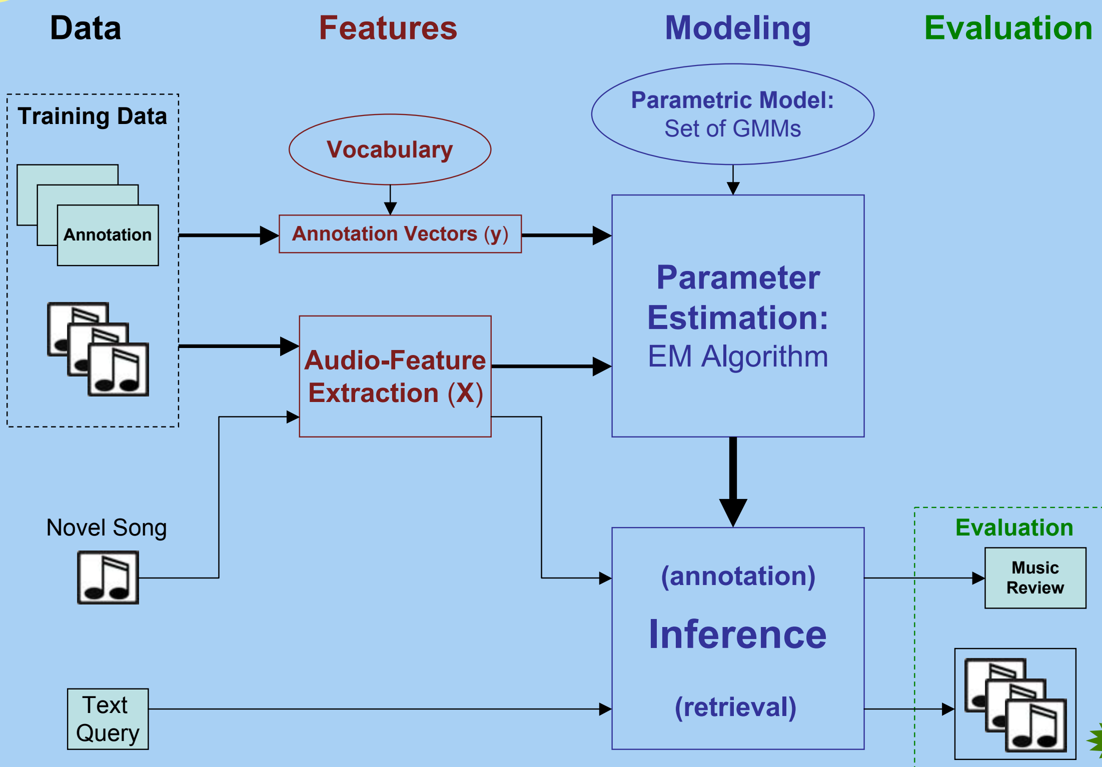
Mean AROC is the average AROC over a large number of queries.

Quantifying Retrieval

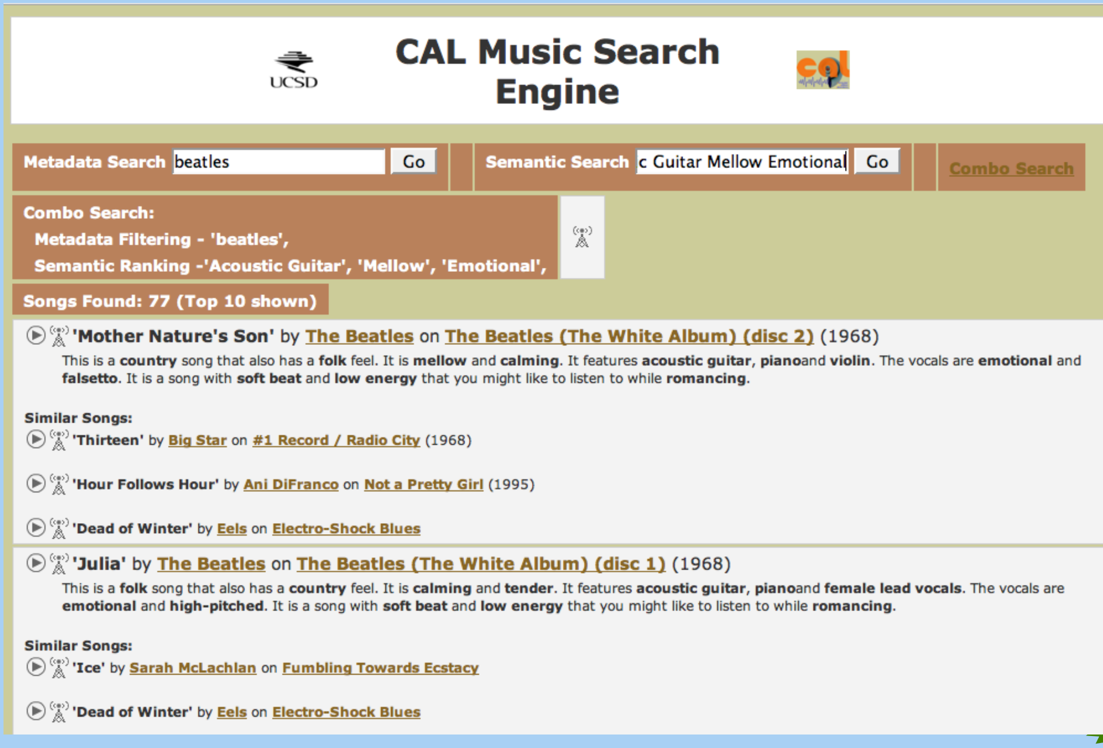
We rank order songs according to KL once for each query.

Model	AROC
Random	0.50
Upper Bound	1.00
Our System - 1 Word	0.71
Our System - 2 Words	0.72
Our System - 3 Words	0.73

System Overview



CAL Music Search Engine



What's on tap…

Building 'Commercial Grade' system

1. Collecting data
   - 'Legally' collecting music
   - Listen Game -> Herd It Game
2. Vocabulary expansion
   - LastFM - 25,000 tags
     - Vocab selection using Sparse CCA - ISMIR 07
   - Web Documents - All words
3. User interface design
   - Natural language music search engine
   - Customizable radio player
4. Automated 'Large Scale' System

What's on tap…

Machine Learning Challenges

1. Derive song similarity
   - Query-by-semantic-example - ICASSP 07, MIREX 07
2. Model correlation between labels
3. Explore discriminative approaches
4. Combine heterogeneous data sources
   - Game Data, Semantic Tags, Web Documents, Popularity Info
5. Focus on individuals / groups rather than population
   - Emotional state of listener

"Talking about music is like dancing about architecture"

- origins unknown

Gert Lanckriet
Computer Audition Lab
UC San Diego

gert@ece.ucsd.edu
cosmal.ucsd.edu/~gert