Latent variable models for audio spectra

Paris Smaragdis
Adobe Systems Inc.
paris@adobe.com
Motivation

- Real sounds are *always* observed within mixtures
- Generic sound models are (semi) blind to mixing
  - Gaussian mixtures
  - Gaussian HMMs
  - Vector Quantization
  - etc …
- Are there models which can gracefully deal with mixing?
Latent variable models for audio

- Low rank generative models for audio
  - Finding hidden underlying structure in spectra
- Family of related models
  - Simple spectral factorization
  - Shift-invariant factorization
  - Entropic and temporal priors
- Various applications and demos
A probabilistic view of spectra

- The magnitude spectrum is a histogram of frequency localized “sound atoms”
  - Each bin describes how much acoustic energy we have at a particular frequency
- And the spectrogram
  - Same thing for time/frequency atoms
- The histogram analogy opens up a door to probabilistic reasoning
  - The histogram is drawn from a distribution describing the sound
  - We can use distribution-based methods on the spectra
Marginals of spectrograms

- The marginals of a 2-D distribution describe the distribution of the two involved variables
  \[ P(f) = \int P(f, t) dt \]
  \[ P(t) = \int P(f, t) df \]

- In a “spectrogram distribution” these variables are time and frequency
- The marginals are the power spectrum and the signal envelope
  - Nothing special here
A single set of marginals is not particularly informative, more info would be better

Let's extend previous concept to employ multiple sets of marginals as opposed to a single set:

\[ P(x) = \prod_{i}^{N} P(x_i) \rightarrow P(x) = \sum_{z} P(z) \prod_{i}^{N} P(x_i \mid z) \]

- \( z \) is a latent variable ("index of marginal sets")
  - Can also be continuous (and messy)
- \( P(z) \) is the prior/weight of the \( z \)-th marginal set
- \( P(x_i \mid z) \) are distributions across the \( i \)-th dimension
Estimating the model distributions

- Using Expectation-Maximization we get:
  - E-step: Contribution of each marginal set to the approximation
    \[
    R(x, z) = \frac{P(z) \prod_{j=1}^{N} P(x_j \mid z)}{\sum_{z'} P(z') \prod_{j=1}^{N} P(x_j \mid z')}
    \]
  - M-step: A weighted marginal estimation using the above factors
    \[
    P(z) = \int P(x) R(x, z) dx
    \]
    \[
    P(x_j \mid z) = \frac{\int \cdots \int P(x) R(x, z) dx_k, \forall k \neq j}{P(z)}
    \]
Simpler than it looks!

- function [w,p,h] = plca2( v, r, ep)

- % Init
- [m,n] = size( v);
- w = rand( m, r);
- h = rand( r, n);
- p = rand( r, 1);

- % Start crunching
- for e = 1:ep

- % Update marginals
- R = v ./ (w*diag(p)*h);
- nw = w .* (R*(diag(p)*h)');
- nh = (diag(p)*h) .* (w'*R);

- % Normalize
- p = sum( nw, 1);
- w = nw .* repmat( 1./p, m, 1);
- h = nh .* repmat( 1./p', 1, n);
- p = p / sum( p);
- end
What does PLCA extract?

- 2-D case with \( z = \{z_1, z_2\} \)
  \[
P(x_1, x_2) = P(z_1)P(x_1 \mid z_1)P(x_2 \mid z_1) + P(z_2)P(x_1 \mid z_2)P(x_2 \mid z_2)
  \]

- \( x_i \) marginals: \( P( x_i \mid z_1 ), P( x_i \mid z_2 ) \)
  - Describe data along the \( x_i \) axis
  - Some kind of \( x_i \) vector codebook

- \( P(z) \)
  - Describes the mixing ratio of the two sets of marginals

- The low rank description is an information bottleneck
Example of time/frequency segmentation

- Simple piano passage
  - Multiple notes
  - Variety in spectral and temporal distributions
  - Can’t be characterized by a single set of marginals
- Extracted marginals
  - Frequency marginals describe the spectra of the notes
  - Time marginals describe their corresponding energy in time
- Doesn’t require isolated notes
- Without supervision we have discovered musical structure!

Paris Smaragdis, Latent variable models for audio spectra
Large scale version

- First six bars of Bach’s fugue XVI in Gm
Looks familiar?

- Can be notated in linear algebra

\[
P(x) = \sum_{z} P(z)P(x_1 \mid z)P(x_2 \mid z) \rightarrow X = P_{x1} \cdot P_{z} \cdot P_{x2}
\]

- Duality with nonnegative basis decomposition
  - marginals ↔ bases/weights
  - nonnegative data ↔ probabilities

- 2D case is numerically equivalent to KL-NMF
  - Priors get sucked into the factors

- The advantage here is the probabilistic nature of PLCA
Still looks familiar!

- **2D model is also known as PLSI/A**
  - Probabilistic Latent Semantic Indexing
  - Probabilistic Latent Semantic Analysis
- **Slight conceptual differences**
  - PLSI is discrete, Poisson assumption
  - In PLCA these are hidden
- **Numerically identical**
- **PLCA advantage is extensibility to multiple dimensions**
  - Allows analysis of multidimensional input structures
And a few more …

- Can also draw links to PCA/SVD/LSI
  - Linear basis models
- Overall advantages of PLCA
  - Non-negativity
    - Crucial for object discovery
  - Generalization to higher dimensions
    - Also nonnegative tensor decompositions
- Probabilistic framework
  - Allows manipulations by priors
  - Integration with learning algorithms
  - Extensible to more exotic forms
Using PLCA to learn and describe sounds

- We can learn frequency marginals from examples of sound classes
  - We thus obtain a spectral dictionary for each sound class
  - This is a unique set for each class of sounds!
- This extraction also works on mixtures!
  - Spectral distributions are, for practical purposes, additive
  - (so is our model)
Source separation of known sound types

- Assumption: Sound mixtures are composed by marginals of the included sounds
  - Which we can learn from examples
- To separate estimate only their proportions in the input:
  \[
P(f, t) = \sum_z P(z)P(f \mid z)P(t \mid z)
  \]
- Then resynthesize using only marginal subset of each sound
- The catch: the sounds must have somewhat dissimilar spectral composition

\[
P(f \mid z) = \begin{cases} 
P_{\text{chimes}}(f \mid z) \\ P_{\text{speech}}(f \mid z) \end{cases}
\]
Source separation with some unknown sounds

- We usually don’t know the frequency marginals for all sounds in a mixture
  - We might only know some
- Complementary learning

\[
P(f,t) = \sum_z P(z)P(f | z)P(t | z)
\]

\[
P(f | z) = \begin{cases} P_{\text{known}}(f | z) & \text{Known/fixed} \\ P_{\text{unknown}}(f | z) & \text{Estimated} \end{cases}
\]

- Unknown sounds are treated as one new set of info we learn online
- Useful in cases where we have unknown/unexpected parts
Special case - denoising

- Very similar to a Wiener filter
  - Instead of a single and rigid noise model we have a dictionary describing the interfering sounds
- Can be done in two ways
  - Have model of noise, extract extras
  - Have model of target, remove extras
- Quality of results depends on how similar the noise is to the target
  - Superior performance for non-stationary noise removal
- Can also use additional temporal and co-occurrence constrains
  - Markovian structure, etc ...

Speech + the beauty of mechanics

- Similar noise
- Wideband noise
- Loosely correlated “noise”
Why not use some other kind of codebook?

- We need to measure the presence of something
  - Therefore our domain is inherently non-negative
- PCA, ICA, LS, etc don’t work
  - The use of cross-cancellation results into non-sensical results
- VQ is not additive
  - Can’t model mixtures
- NMF is ok
  - Caveats coming up later
Measuring the presence of marginals

- In PLCA there is a 1-to-1 mapping between frequency and time marginals
  - Spectrum $P(f | z_i)$ is modulated by $P(z_i) P(t | z_i)$
- The time marginals indicate amount of presence of frequency marginals across the input’s timeline
- The likelihood of $P(f | z_i)$ at time $t$ is:

$$\sum \sum P(z_i) P(t | z_i) P^{(f,t)}$$
Sound recognition in mixtures

- Use known marginals to estimate presence of these sounds in a mixture

\[ P(f,t) = \sum_{z} P(z) \begin{bmatrix} P_{\text{shaker}}(f \mid z) \\ P_{\text{cymbals}}(f \mid z) \\ P_{\text{jingles}}(f \mid z) \\ P_{\text{pig}}(f \mid z) \end{bmatrix} \cdot \begin{bmatrix} P_{\text{shaker}}(t \mid z) \\ P_{\text{cymbals}}(t \mid z) \\ P_{\text{jingles}}(t \mid z) \\ P_{\text{pig}}(t \mid z) \end{bmatrix} \]

- We can now do meta-learning
- E.g. HMM extension for concurrent speaker recognition
  - State model is PLCA
  - \(~87\%\) recognition on DG 0dB speech separation challenge data
  - No factorial decoding needed!
Temporal models

- Enforce a sense of time order
  - E.g. using a dynamical model or time warping
- Takes into account articulatory details of sounds
  - Allows separation of similar sounds with different temporal characteristics
  - E.g. spoken /s/ vs wideband noise
- HMM extension example
  - PLCA state model
  - ~87% recognition accuracy on different gender 0dB speech mixtures
  - No need for factorial models
Audio superresolution

- Data imputation using PLCA
- Knowing how things should sound
  - Analyze bandlimited input using known marginals at available frequencies only
  - Reconstruct using full bandwidth example marginals
- “Resampling by example”
Missing data imputation without training

Large gap missing data input

Recovered output

Random mask missing data input

Recovered output
Audio-visual example

- Input with correlated images and pictures
  - Results into multimodal features
  - Reveals key/note relationship
Shift invariant form

- PLCA with shift invariance:
  \[ P(x) = \sum_i P(z_i) \int P_K(w, t | z_i) P_I(y - t | z_i) \]
- Estimates kernel distributions \( P_K \) which are placed by impulse distributions \( P_I \)
  - Dual to a convolutive basis decomposition
- Easy to derive parameters
  - Straightforward EM
  - Can be sped up with FFTs
- Some complications due to overcompleteness
  - What’s a kernel and what’s an impulse?
  - Convolution is commutative
Sparsity constraints

- Entropic manipulations on the extracted data
  - Make any distribution flatter or sparser
- Change to MAP estimation, use an entropic prior:
  - $P(q) = e^{-\beta H(q)}$
  - $\beta$ is a weight, the rest introduces a bias towards a high or low entropy estimate
- No “correct” solution
  - Representation is often overcomplete
  - Results are biased to suit needs
- Sparsity on $P(z)$ helps in determining dimensionality of problem
PLCA vs Shift Invariant PLCA on audio

- Finding *repeating chunks* instead of *slices*
- Extracts short-time temporal structure
  - More informative description
  - Also less expressive
- Applicable to all previous examples
  - Separation, recognition, denoising, etc.
- Also probabilistic and easy to plug into meta-learning algorithms

Input spectrogram

PLCA frequency marginals

Shift-invariant kernels
Shift-invariant PLCA on speech

- On speech, shift-invariant PLCA discovers phones (parts of phonemes)

- These can be used do all of the previous work on separation much more robustly since we now take temporal aspects into account
PLCA for pitch tracking

- Constant-Q transform makes transposition a shift in frequency
  - Use shift-invariant analysis on that axis too
- Kernel will be source spectrum
- Impulse will be pitch probability function
- Doesn’t matter if the source is harmonic or not!
Extracting multiple pitch tracks

- Extracting mixture pitch tracks
  - Do constant-Q transforms
  - Use shiftable components
  - We still have additivity
- In simple model we obtain pitch tracks and sources
  - Assumption of semi-constant spectral shape for each source
- Can recover unpitched sources
  - No harmonicity assumption
- Two-part example
  - Voice + bell mixture
  - “Weights” provide a rough pitch track
  - Inharmonicity of bell is not an issue
  - Both spectral profiles are extracted
Applications on images

- Images can be interpreted as distributions as well
  - \{x,y,r,g,b\} photon counter
- Shift-invariant PLCA finds repeating patterns
- Extensions with other types of kernel variations (rotation, scale, …)
Enough with the toy examples

- Getting things out to the real world
  - Writing papers is fun
  - Making things that work is the real challenge though!
- Lots of applications are there for the taking
  - Audio is an immensely informative mode
  - Audio has some distinct advantages
    - High entropy, little data
      - i.e. cheap processing, transmission, storage
- Can we make the above work for real?
Video Content Analysis

- Audio is a strong cue for detecting various events
- Classify sounds to perform semantic analysis on video
  - Specific subclasses for type of broadcast (e.g. for news we use male and female speech, for sports use cheering, etc)
- Result used for content analysis
  - Clustering videos
  - Finding sections
  - Scene searching
  - Commercial detection
    - Commercials “sound different”
- Build in high-end Mitsubishi PVRs, upcoming TV sets and “HDTV cell phones”

Hard computer vision problems
Security Applications

- Based on the sound recognition work we can classify sounds in various environments
- Traffic surveillance
  - Detect interesting sounding cases
  - Impacts, tire screeching, honking
- Elevator security
  - Normal speech, excited speech, footsteps, thumps, door open/close, screams
- When detecting suspicious sounds we can raise an alert
  - Mid to high 90s% accuracy
- Very cheap/efficient system

Examples of actual detected “interesting” parts

Elevators are a dark environment with poor visual analysis prospects
Audio layer editing

- Original drum loop
- Extracted layers
- No tambourine
- No congas
- Congas!
- Remixer
- Selective pitch shifting

Piano + Soprano
- Soprano layer
- Piano layer
- Remixed layers

Paris Smaragdis, Latent variable models for audio spectra
Wrap up

- Latent variable models for spectra
- Important properties
  - Non-negativity is important
  - A probabilistic handle on spectral factorizations
- Extensible in various ways
  - Priors on sparsity, temporal continuity, etc
- A competitive alternative to standard models
  - Additivity property makes this model appropriate for dealing with mixtures of sounds
- Seem to work fine in real life applications!