11-755 Machine Learning for Signal Processing

Latent Variable Models and Signal Separation

Class 9. 29 Sep 2011

The Engineer and the Musician

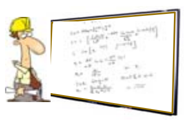
Once upon a time a rich potentate discovered a previously unknown recording of a beautiful piece of music. Unfortunately it was badly damaged.

He greatly wanted to find out what it would sound like if it were not.

So he hired an engineer and a musician to solve the problem..

The Engineer and the Musician

The engineer worked for many years. He spent much money and published many papers.



Finally he had a somewhat scratchy restoration of the music..

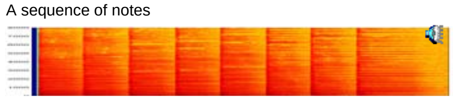
The musician listened to the music carefully for a day, transcribed it, broke out his trusty keyboard and replicated the music.

The Prize

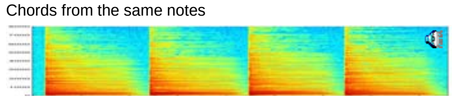
Who do you think won the princess?

Sounds – an example

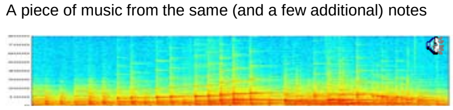
- A sequence of notes



- Chords from the same notes

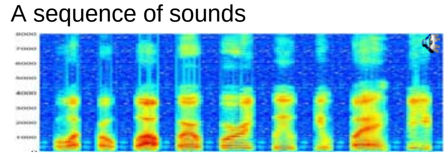


- A piece of music from the same (and a few additional) notes

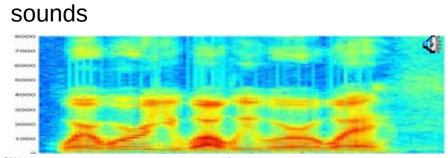


Sounds – an example

- A sequence of sounds



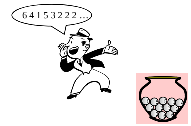
- A proper speech utterance from the same sounds



Template Sounds Combine to Form a Signal

- The individual component sounds "combine" to form the final complex sounds that we perceive
  - Notes form music
  - Phoneme-like structures combine in utterances
  - Component sounds – notes, phonemes – too are complex
- Sound in general is composed of such "building blocks" or themes
  - Our definition of a building block: the entire structure occurs repeatedly in the process of forming the signal
- Goal: To learn these building blocks automatically, from analysis of data

Urns and balls

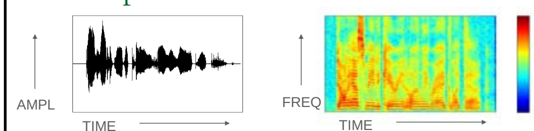


- An urn has many balls
- Each ball has a number marked on it
  - Multiple balls may have the same number
- A "picker" draws balls at random..
- This is a multinomial

Signal Separation with the Urn model

- What does the probability of drawing balls from Urns have to do with sounds?
  - Or Images?
- We shall see..

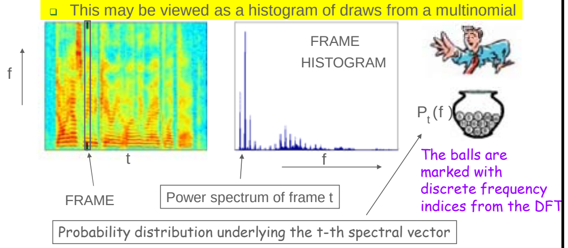
The representation



- We represent signals spectrographically
  - Sequence of magnitude spectral vectors estimated from (overlapping) segments of signal
  - Computed using the short-time Fourier transform
  - Note: Only retaining the magnitude of the STFT for our operations
  - We will, however need the phase later for conversion to a signal

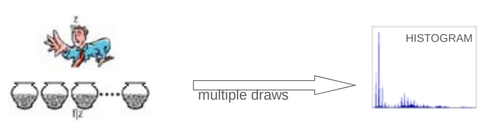
A Multinomial Model for Spectra

- A magnitude spectral vector obtained from a DFT represents spectral magnitude against discrete frequencies
  - This may be viewed as a histogram of draws from a multinomial

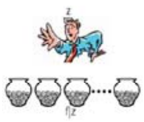


A more complex model

- A "picker" has multiple urns
- In each draw he first selects an urn, and then a ball from the urn
  - Overall probability of drawing *f* is a *mixture multinomial*
    - Since several multinomials (urns) are combined
  - Two aspects – the probability with which he selects any urn, and the probability of frequencies with the urns

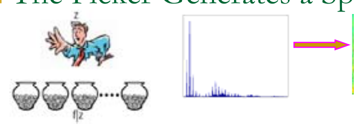


The Picker Generates a Spectrogram



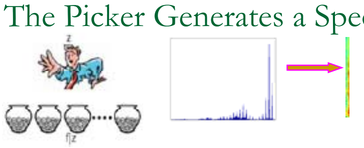
- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to some probability $P_1(z)$
- And so on, until he has constructed the entire spectrogram

The Picker Generates a Spectrogram



- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to some probability $P_1(z)$
- And so on, until he has constructed the entire spectrogram

The Picker Generates a Spectrogram

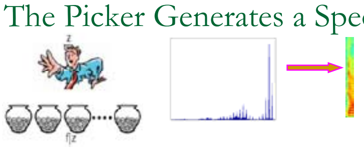


- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to some probability $P_1(z)$
- And so on, until he has constructed the entire spectrogram

The Picker Generates a Spectrogram

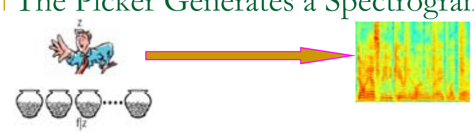
- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to some probability $P_1(z)$
- And so on, until he has constructed the entire spectrogram

The Picker Generates a Spectrogram



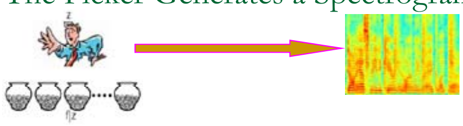
- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to some probability $P_1(z)$
- And so on, until he has constructed the entire spectrogram

The Picker Generates a Spectrogram



- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to some probability $P_1(z)$
- And so on, until he has constructed the entire spectrogram
  - The number of draws in each frame represents the rms energy in that frame

The Picker Generates a Spectrogram



- The URNS are the same for every frame
  - These are the ***component multinomials*** or ***bases*** for the source that generated the signal
- The only difference between frames is the probability with which he selects the urns

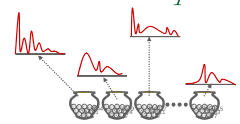
$$P_t(f) = \sum_z P_t(z) P(f \mid z)$$

Frame-specific spectral distribution

SOURCE specific bases

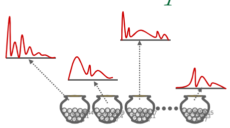
Frame(time) specific mixture weight

Spectral View of *Component* Multinomials



- Each component multinomial (urn) is actually a normalized histogram over frequencies $P(f \mid z)$
  - I.e. a spectrum
- Component multinomials represent latent spectral structures (bases) for the given sound source
- The spectrum for *every* analysis frame is explained as an additive combination of these latent spectral structures

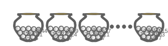
Spectral View of *Component* Multinomials



- By "learning" the mixture multinomial model for any sound source we "discover" these latent spectral structures for the source
- The model can be learnt from spectrograms of a small amount of audio from the source using the EM algorithm

EM learning of bases

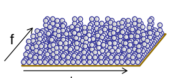
- Initialize bases
  - P(f|z) for all z, for all f
    - Must decide on the number of urns



- For each frame
  - Initialize $P_t(z)$

Learning the Bases

- Simple EM solution
  - Except bases are learned from *all* frames

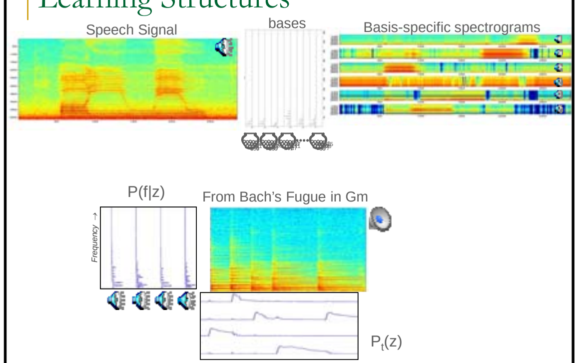


$$P_t(z \mid f) = \frac{P_t(z) P(f \mid z)}{\sum_{z'} P_t(z') P(f \mid z')}$$ fragmentation

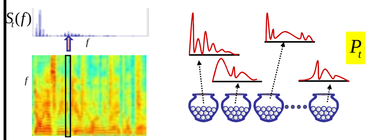
$$P_t(z) = \frac{\sum_f P_t(z \mid f) S_t(f)}{\sum_{z'} \sum_f P_t(z' \mid f) S_t(f)}$$ counting

$$P(f \mid z) = \frac{\sum_t P_t(z \mid f) S_t(f)}{\sum_{f'} \sum_t P_t(z \mid f') S_t(f')}$$ The "Basis" distribution

Learning Structures



Given Bases Find Composition



$$P_t(f) = \sum_z P_t(z) P(f \mid z)$$

- Iterative process:
  - Compute a posteriori probability of the z^{th} topic for each frequency f in the t-th spectrum

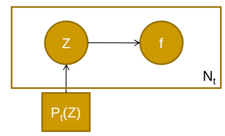
$$P_t(z \mid f) = \frac{P_t(z) P(f \mid z)}{\sum_{z'} P_t(z') P(f \mid z')}$$

  - Compute mixture weight of z^{th} basis

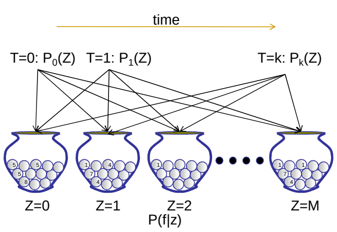
$$P_t(z) = \frac{\sum_f P(z \mid f) S_t(f)}{\sum_{z'} \sum_f P(z' \mid f) S_t(f)}$$

Bag of Frequencies vs. Bag of Spectrograms

- The PLCA model described is a "bag of frequencies" model
  - Similar to "bag of words"
- Composes spectrogram one frame at a time
  - Contribution of bases to a frame does not affect other frames
- Random Variables:
  - Frequency
  - Possibly also the total number of draws in a frame

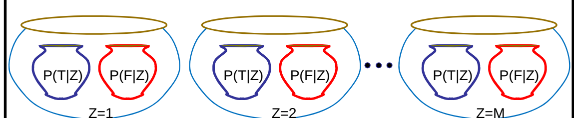


Bag of Frequencies PLCA model



- Bases are simple distributions over frequencies
- Manner of selection of urns/components varies from analysis frame to analysis frame

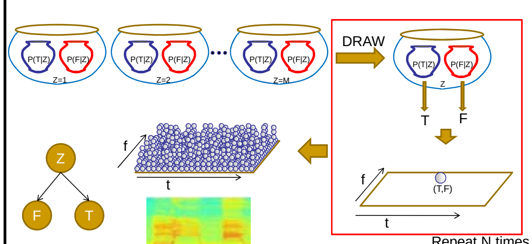
Bag of Spectrograms PLCA Model



- Compose the entire spectrogram all at once
- Complex "super pots" include two sub pots
  - One pot has a distribution over frequencies: these are our bases
  - The second has a distribution over time
- Each draw:
  - Select a superpot
  - Draw "F" from frequency pot
  - Draw "T" from time pot
  - Increment histogram at (T,F)

$$P(t, f) = \sum_z P(z) P(t \mid z) P(f \mid z)$$

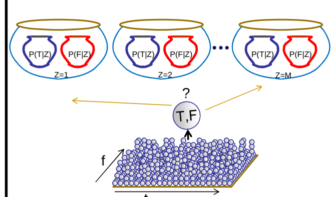
The bag of spectrograms



$$P(t, f) = \sum_z P(z) P(t \mid z) P(f \mid z)$$

- Drawing procedure
  - Fundamentally equivalent to bag of frequencies model
    - With some minor differences in estimation

Estimating the bag of spectrograms



$$P(t, f) = \sum_z P(z) P(t \mid z) P(f \mid z)$$

$$P(z \mid t, f) = \frac{P(z) P(f \mid z) P(t \mid z)}{\sum_{z'} P(z') P(f \mid z') P(t \mid z')}$$

$$P(z) = \frac{\sum_t \sum_f P(z \mid t, f) S_t(f)}{\sum_{z'} \sum_t \sum_f P(z' \mid t, f) S_t(f)}$$

$$P(f \mid z) = \frac{\sum_t P(z \mid t, f) S_t(f)}{\sum_{f'} \sum_t P(z \mid t, f') S_t(f')}$$

$$P(t \mid z) = \frac{\sum_f P(z \mid t, f) S_t(f)}{\sum_{t'} \sum_f P(z \mid t', f) S_{t'}(f)}$$

- EM update rules
  - Can learn all parameters
  - Can learn P(T|Z) and P(Z) only given P(f|Z)
  - Can learn only P(Z)

Bag of frequencies vs. bag of spectrograms

- Fundamentally equivalent
- Difference in estimation
  - Bag of spectrograms: For a given total N and P(Z), the total "energy" assigned to a basis is determined
    - increasing its energy at one time will necessarily decrease its energy elsewhere
    - No such constraint for bag of frequencies
      - More unconstrained
  - Can also be used to assign temporal patterns for components
- Bag of frequencies more amenable to imposition of *a priori* distributions
- Bag of spectrograms a more natural fit for other models

The PLCA Tensor Model

- The bag of spectrograms can be extended to multivariate data

$$P(a,b,...c) = \sum_z P(z)P(a|z)P(b|z)...P(c|z)$$

- EM update rules are essentially identical to bivariate case

How meaningful are these structures

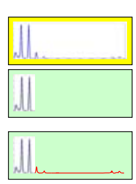
- If bases capture data structure they must
  - Allow prediction of data
    - **Hearing only the low-frequency components of a note, we can still know the note**
    - **Which means we can predict its higher frequencies**
  - Be resolvable in complex sounds
    - Must be able to pull them out of complex mixtures
      - **Denoising**
      - **Signal Separation from Monaural Recordings**

The musician vs. the signal processor

- Some badly damaged music is given to a signal processing whiz and a musician
  - They must "repair" it. What do they do?
- Signal processing :
  - Invents many complex algorithms
  - Writes proposals for government grants
  - Spends $1000,000
  - Develops an algorithm that results in less scratchy sounding music
- Musician:
  - Listens to the music and transcribes it
  - Plays it out on his keyboard/piano

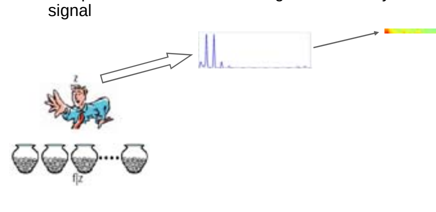
Prediction

- **Bandwidth Expansion**
  - Problem: A given speech signal only has frequencies in the 300Hz-3.5Khz range
    - Telephone quality speech
  - Can we estimate the rest of the frequencies
- The full basis is known
- The presence of the basis is identified from the observation of a part of it
- The obscured remaining spectral pattern can be guessed



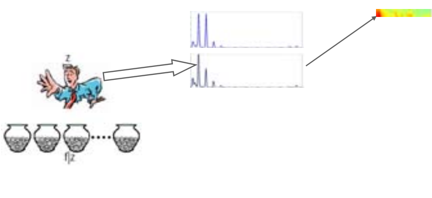
Bandwidth Expansion

- The picker has drawn the histograms for every frame in the signal



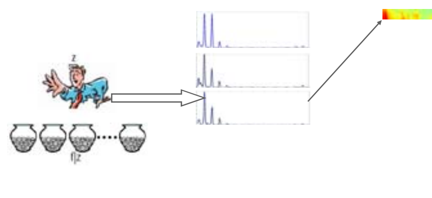
Bandwidth Expansion

- The picker has drawn the histograms for every frame in the signal



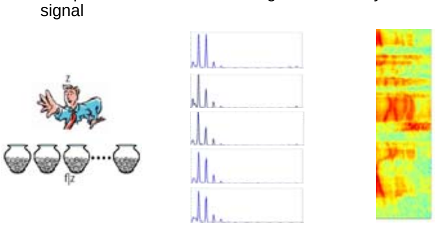
Bandwidth Expansion

- The picker has drawn the histograms for every frame in the signal



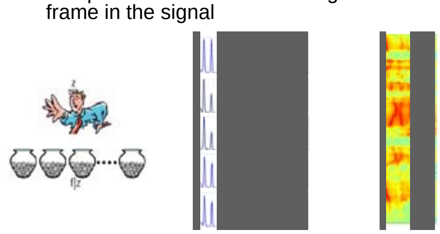
Bandwidth Expansion

- The picker has drawn the histograms for every frame in the signal



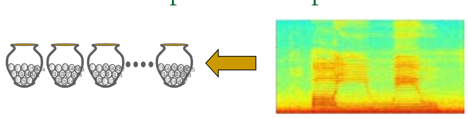
Bandwidth Expansion

- The picker has drawn the histograms for every frame in the signal



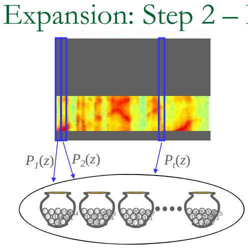
- However, we are only able to observe the number of draws of some frequencies and not the others
- We must estimate the number of draws of the unseen frequencies

Bandwidth Expansion: Step 1 – Learning



- From a collection of ***full-bandwidth*** training data that are similar to the bandwidth-reduced data, learn spectral bases
  - Using the procedure described earlier

Bandwidth Expansion: Step 2 – Estimation



- Using *only the observed frequencies* in the bandwidth-reduced data, estimate mixture weights for the bases learned in step 1.

Step 2

- Iterative process:
  - Compute a posteriori probability of the z^{th} urn for the speaker for each *f*

$$P_t(z|f)=\frac{P_t(z)P(f|z)}{\sum_{z'}P_t(z')P(f|z')}$$

  - Compute mixture weight of z^{th} urn for each frame *t*

$$P_t(z)=\frac{\sum_{f\in(\text{observed frequencies})}P_t(z|f)S_t(f)}{\sum_{z'}\sum_{f\in(\text{observed frequencies})}P_t(z'|f)S_t(f)}$$

  - P(f|z) was obtained from training data and will not be reestimated

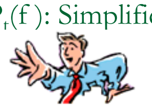
Step 3 and Step 4

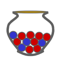
- Compose the complete probability distribution for each frame, using the mixture weights estimated in Step 2

$$P_t(f)=\sum_z P_t(z)P(f|z)$$

- Note that we are using mixture weights estimated from the reduced set of observed frequencies
  - This also gives us estimates of the probabilities of the *unobserved* frequencies
- Use the complete probability distribution $P_t(f)$ to predict the unobserved frequencies!

Predicting from $P_t(f)$: Simplified Example





- A single Urn with only red and blue balls
- Given that out an unknown number of draws, exactly *m* were red, how many were blue?
- One Simple solution:
  - Total number of draws N = m / P(red)
  - The number of tails drawn = N*P(blue)
  - Actual multinomial solution is only slightly more complex

The inverse multinomial

- Given P(Z) for all bases
- Observed n_1, n_2 .. n_k
- What is n_{k+1}, n_{k+2}…

$$P(n_{k+1},n_{k+2},\ldots)=\frac{\Gamma\left(N_o+\sum_{f>k}n_f\right)}{\Gamma(N_o)\Gamma\left(\sum_{f>k}n_f\right)}P_o\prod_{f>k}P(f)^{n_f}$$

- N_o is the total number of observed counts
  - $n_1 + n_2 + \ldots$
- P_o is the total probability of observed events
  - $P(f_1) + P(f_2) + \ldots$

Estimating unobserved frequencies

- Expected value of the number of draws:

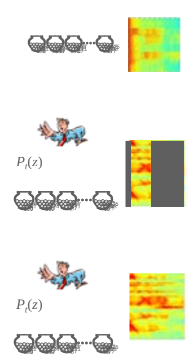
$$\hat{N}_t=\frac{\sum_{f\in(\text{observed frequencies})}S_t(f)}{\sum_{f\in(\text{observed frequencies})}P_t(f)}$$

- Estimated spectrum in unobserved frequencies

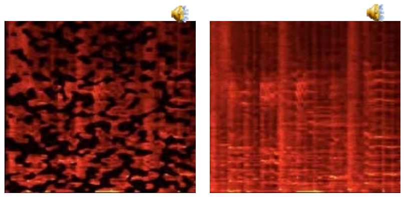
$$\hat{S}_t(f)=\hat{N}_tP_t(f)$$

Overall Solution

- Learn the “urns” for the signal source from broadband training data
- For each frame of the reduced bandwidth test utterance, find mixture weights for the urns
  - Ignore (marginalize) the unseen frequencies
- Given the complete mixture multinomial distribution for each frame, estimate spectrum (histogram) at unseen frequencies



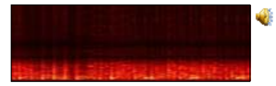
Prediction of Audio



- Some frequency components are missing (left panel)
- We know the bases P(f|z)
  - But not the mixture weights for any particular spectral frame
- We must “fill in” the hole in the image
  - To obtain the one to the right
  - Easy to do – as explained

A more fun example

•Reduced BW data



•Bases learned from this

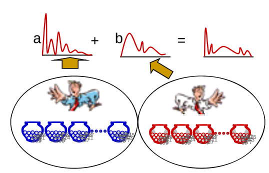
•Bandwidth expanded version

Signal Separation from Monaural Recordings

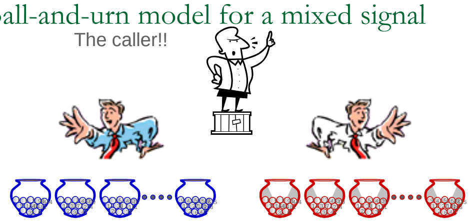
- The problem:
  - Multiple sources are producing sound simultaneously
  - The combined signals are recorded over a single microphone
  - The goal is to selectively separate out the signal for a target source in the mixture
    - Or at least to enhance the signals from a selected source

Problem Specification

- The mixed signal contains components from multiple sources
- Each source has its own “bases”
- In each frame
  - Each source draws from its own collection of bases to compose a spectrum
    - Bases are selected with a frame specific mixture weight
  - The overall spectrum is a mixture of the spectra of individual sources
    - I.e. a histogram combining draws from both sources
- Underlying model: Spectra are histograms over frequencies



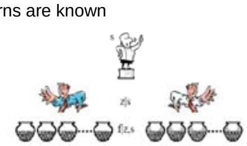
Ball-and-urn model for a mixed signal



- Each sound source is represented by its own picker and urns
  - Urns represent the distinctive spectral structures for that source
  - **Assumed to be known beforehand** (learned from some separate training data)
- The caller selects a picker at random
  - The picker selects an urn randomly and draws a ball
  - The caller calls out the frequency on the ball
- A spectrum is a histogram of frequencies called out
  - The total number of draws of any frequency includes contributions from *both* sources

Separating the sources

- Goal: Estimate number of draws from each source
  - The probability distribution for the mixed signal is a linear combination of the distribution of the individual sources
  - The individual distributions are mixture multinomials
  - And the urns are known

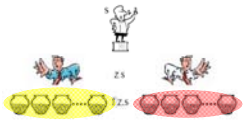


$$P_t(f) = P_t(s_1)P_t(f \mid s_1) + P_t(s_2)P_t(f \mid s_2)$$

$$\boxed{P_t(f) = P_t(s_1)\sum_z P_t(z \mid s_1)P(f \mid z, s_1) + P_t(s_2)\sum_z P_t(z \mid s_1)P(f \mid z, s_2)}$$

Separating the sources

- Goal: Estimate number of draws from each source
  - The probability distribution for the mixed signal is a linear combination of the distribution of the individual sources
  - The individual distributions are mixture multinomials
  - And the urns are known

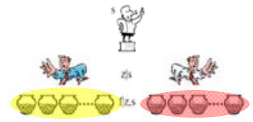


$$P_t(f) = P_t(s_1)P_t(f \mid s_1) + P_t(s_2)P_t(f \mid s_2)$$

$$P_t(f) = P_t(s_1)\sum_z P_t(z \mid s_1)P(f \mid z, s_1) + P_t(s_2)\sum_z P_t(z \mid s_1)P(f \mid z, s_2)$$

Separating the sources

- Goal: Estimate number of draws from each source
  - The probability distribution for the mixed signal is a linear combination of the distribution of the individual sources
  - The individual distributions are mixture multinomials
  - And the urns are known
  - **Estimate remaining terms using EM**



$$P_t(f) = P_t(s_1)P_t(f \mid s_1) + P_t(s_2)P_t(f \mid s_2)$$

$$P_t(f) = P_t(s_1)\sum_z P_t(z \mid s_1)P(f \mid z, s_1) + P_t(s_2)\sum_z P_t(z \mid s_1)P(f \mid z, s_2)$$

Algorithm

- For each frame:
  - Initialize $P_t(s)$
    - The fraction of balls obtained from source s
    - Alternately, the fraction of energy in that frame from source s
  - Initialize $P_t(z|s)$
    - The mixture weights of the urns in frame *t* for source s
  - Reestimate the above two iteratively
- Note: P(f|z,s) is not frame dependent
  - It is also not re-estimated
  - Since it is assumed to have been learned from separately obtained unmixed training data for the source

Iterative algorithm

- Iterative process:
  - Compute a posteriori probability of the combination of speaker s and the z[th] urn for each speaker for each f

$$P_t(s,z \mid f) = \frac{P_t(s)P_t(z \mid s)P(f \mid z,s)}{\sum_{s'} P_t(s')\sum_{z'} P_t(z' \mid s')P(f \mid z',s')}$$

  - Compute the a priori weight of speaker s

$$P_t(s) = \frac{\sum_z \sum_f P_t(s,z \mid f)S_t(f)}{\sum_{s'}\sum_{z'}\sum_f P_t(s',z' \mid f)S_t(f)}$$

  - Compute mixture weight of z[th] urn for speaker s

$$P_t(z \mid s) = \frac{\sum_f P_t(s,z \mid f)S_t(f)}{\sum_{z'}\sum_f P_t(s',z' \mid f)S_t(f)}$$

What is $P_t(s,z|f)$

- Compute how each ball (frequency) is split between the urns of the various sources
- The ball is first split between the sources

$$P_t(s \mid f) = \frac{P_t(s)}{\sum_{s'} P_t(s')}$$

- The fraction of the ball attributed to any source s is split between its urns:

$$P_t(z \mid s, f) = \frac{P_t(z \mid s)P(f \mid z,s)}{\sum_{z'} P_t(z' \mid s)P(f \mid z',s)}$$

- The portion attributed to any urn of any source is a product of the two

$$P_t(s,z \mid f) = \frac{P_t(s)P_t(z \mid s)P(f \mid z,s)}{\sum_{s'} P_t(s')\sum_{z'} P_t(z' \mid s')P(f \mid z',s')}$$

Reestimation

- The reestimate of source weights is simply the proportion of all balls that was attributed to the sources

$$P_t(s) = \frac{\sum_z \sum_f P_t(s,z \mid f)S_t(f)}{\sum_{s'}\sum_{z'}\sum_f P_t(s',z' \mid f)S_t(f)}$$

- The reestimate of mixture weights is the proportion of all balls attributed to each urn

$$P_t(z \mid s) = \frac{\sum_f P_t(s,z \mid f)S_t(f)}{\sum_{z'}\sum_f P_t(s',z' \mid f)S_t(f)}$$

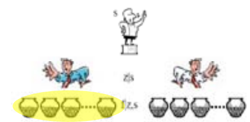
Separating the Sources

- For each frame:
- Given
  - $S_t(f)$ – The spectrum at frequency f of the mixed signal
- Estimate
  - $S_{t,i}(f)$ – The spectrum of the separated signal for the i-th source at frequency f
- A simple maximum a posteriori estimator

$$\hat{S}_{t,i}(f) = S_t(f) \sum_z P_t(z,s \mid f)$$

If we have only have bases for one source?

- Only the bases for one of the two sources is given
  - Or, more generally, for N-1 of N sources

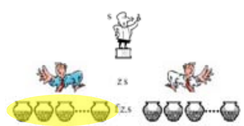


$$P_t(f) = P_t(s_1)P_t(f \mid s_1) + P_t(s_2)P_t(f \mid s_2)$$

$$P_t(f) = P_t(s_1)\sum_z P_t(z \mid s_1)P(f \mid z,s_1) + P_t(s_2)\sum_z P_t(z \mid s_1)P(f \mid z,s_2)$$

If we have only have bases for one source?

- Only the bases for one of the two sources is given
  - Or, more generally, for N-1 of N sources
  - The unknown bases for the remaining source must also be estimated!



$$P_t(f) = P_t(s_1)P_t(f \mid s_1) + P_t(s_2)P_t(f \mid s_2)$$

$$P_t(f) = P_t(s_1)\sum_z P_t(z \mid s_1)P(f \mid z,s_1) + P_t(s_2)\sum_z P_t(z \mid s_1)P(f \mid z,s_2)$$

Partial information: bases for one source unknown

- P(f|z,s) must be initialized for the additional source
- Estimation procedure now estimates bases along with mixture weights and source probabilities
  - From the ***mixed signal itself***
- The final separation is done as before

Iterative algorithm

- Iterative process:
  - Compute a posteriori probability of the combination of speaker s and the z^{th} urn for the speaker for each f

$$P_t(s,z \mid f) = \frac{P_t(s)P_t(z \mid s)P(f \mid z,s)}{\sum_{s'} P_t(s')\sum_{z'} P_t(z' \mid s')P(f \mid z',s')}$$

  - Compute the a priori weight of speaker s and mixture

$$P_t(s) = \frac{\sum_z \sum_f P_t(s,z \mid f)S_t(f)}{\sum_{s'}\sum_{z'}\sum_f P_t(s',z' \mid f)S_t(f)} \qquad P_t(z \mid s) = \frac{\sum_f P_t(s,z \mid f)S_t(f)}{\sum_{z'}\sum_f P_t(s',z' \mid f)S_t(f)}$$

  - Compute unknown bases

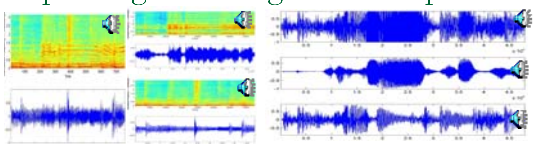
$$P(f \mid z,s) = \frac{\sum_t P_t(s,z \mid f)S_t(f)}{\sum_{f'}\sum_t P_t(s,z \mid f')S_t(f')}$$

Partial information: bases for one source unknown

- P(f|z,s) must be initialized for the additional source
- Estimation procedure now estimates bases along with mixture weights and source probabilities
  - From the ***mixed signal itself***
- The final separation is done as before

$$\hat{S}_{t,i}(f) = S_t(f) \sum_z P_t(z,s \mid f)$$

Separating Mixed Signals: Examples



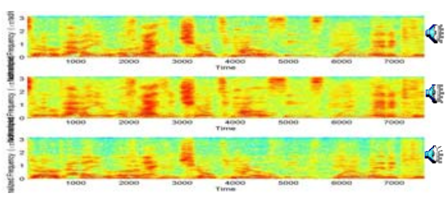
- “Raise my rent” by David Gilmour
- Background music “bases” learnt from 5-seconds of music-only segments within the song
- Lead guitar “bases” bases learnt from the rest of the song

- Norah Jones singing “Sunrise”
- A more difficult problem:
  - Original audio clipped!
- Background music bases learnt from 5 seconds of music-only segments

Where it works

- When the spectral structures of the two sound sources are distinct
  - Don’t look much like one another
  - E.g. Vocals and music
  - E.g. Lead guitar and music
- Not as effective when the sources are similar
  - Voice on voice

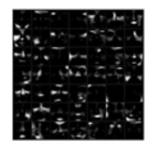
Separate overlapping speech



- Bases for both speakers learnt from 5 second recordings of individual speakers
- Shows improvement of about 5dB in Speaker-to-Speaker ratio for both speakers
  - Improvements are worse for same-gender mixtures

How about non-speech data

19x19 images = 361 dimensional vectors



- We can use the same model to represent other data
- Images:
  - Every face in a collection is a histogram
  - Each histogram is composed from a mixture of a fixed number of multinomials
    - All faces are composed from the same multinomials, but the manner in which the multinomials are selected differs from face to face
  - Each component multinomial is also an image
    - And can be learned from a collection of faces
- Component multinomials are observed to be *parts of faces*