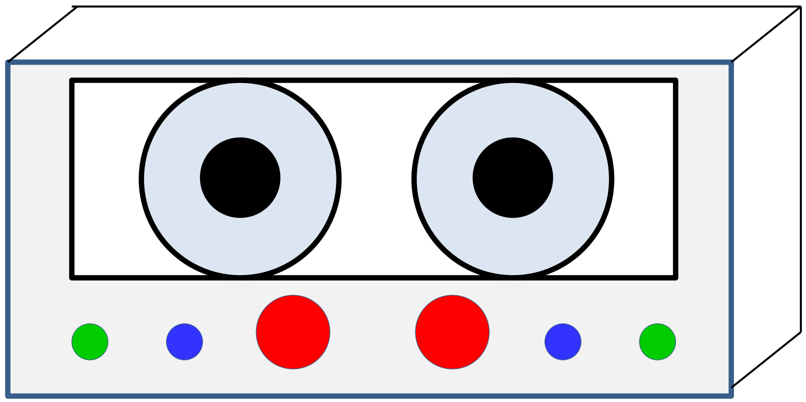
Machine Learning for Signal Processing
Latent Variable Models and Signal Separation

Bhiksha Raj

Class 13. 15 Oct 2013

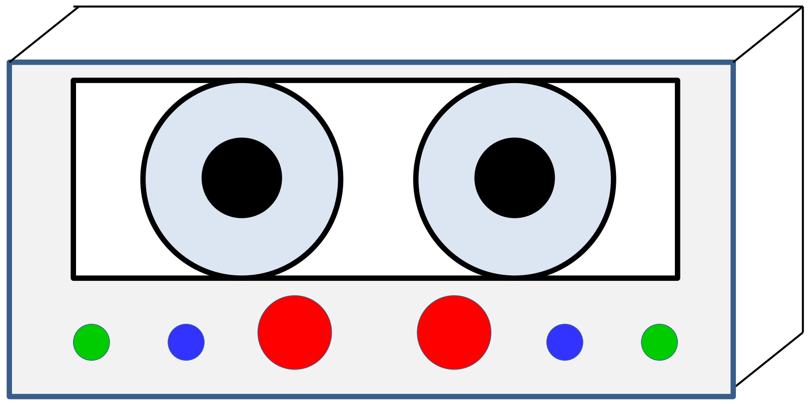
The Great Automatic Grammatinator



IT IT WWAS A AS A BRDAIGRHK T COLAD ND STODARY MY IN NIAPGRHTIL

- The great automatic grammatinator is working hard..

The Great Automatic Grammatinator

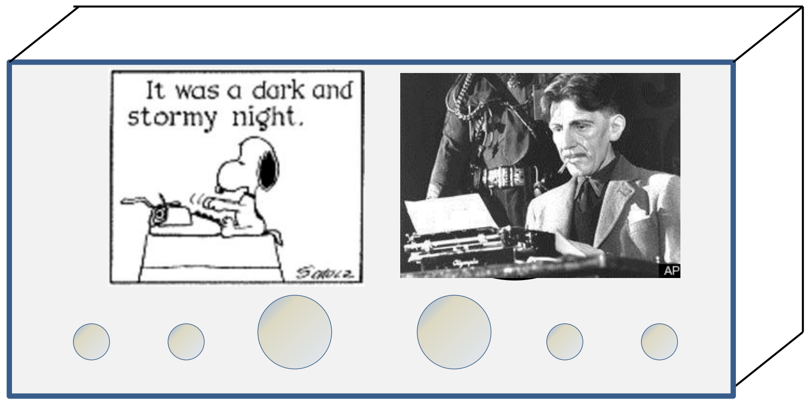


Used with permission of Management Pocketbooks

IT IT WWAS A AS A BRDAIGRHK T COL AD ND STODARY MY IN NIAPGRHTIL

- The great automatic grammatinator is working hard..
  - But what is it writing?

The Secret of the Great Automatic Grammatinator



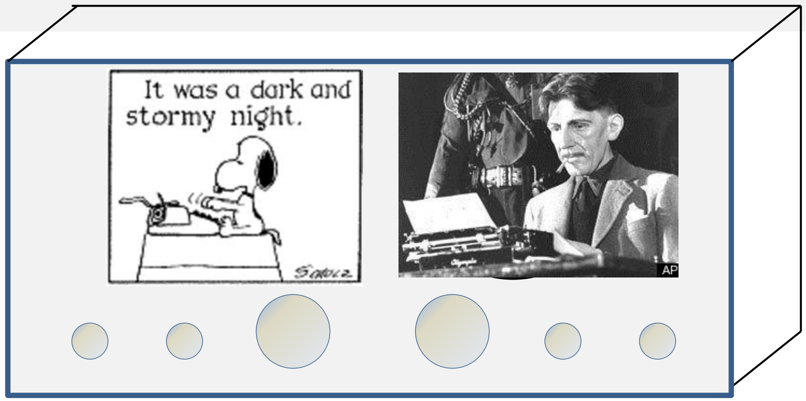
IT WAS A DARK AND STORMY NIGHT…

IT WAS A BRIGHT COLD DAY IN APRIL AND THE CLOCKS WERE STRIKING THIRTEEN …

- The secret of the Grammatinator

The Notion of *Latent* Structure

IT IT WWAS A AS A BRDAIGRHK T COLAD ND STODARY MY IN NIAPGRHTIL



IT WAS A DARK AND STORMY NIGHT…

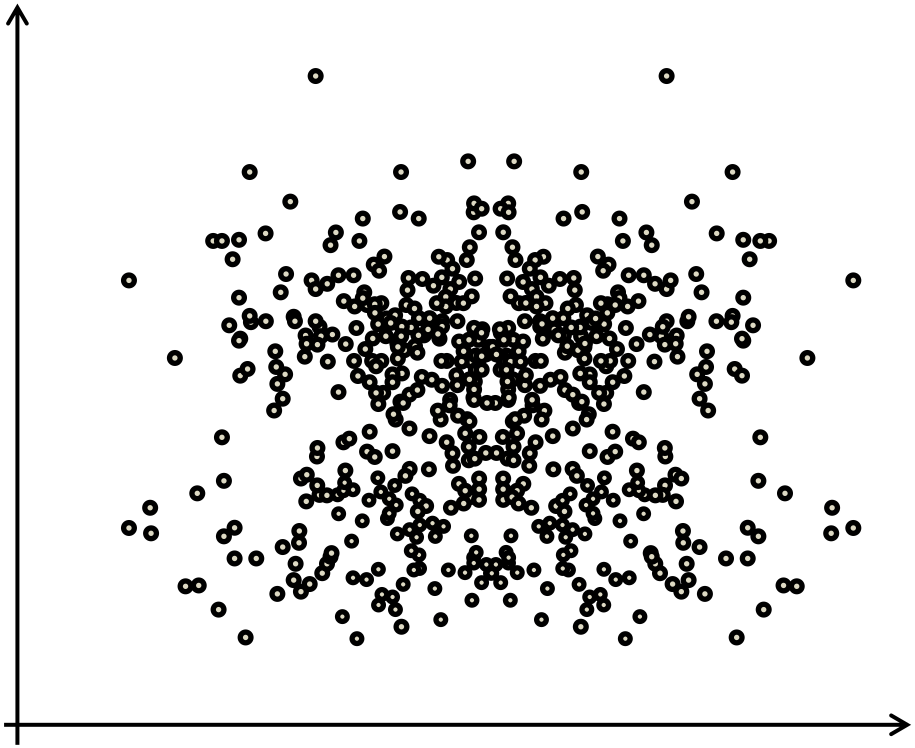
IT WAS A BRIGHT COLD DAY IN APRIL …

- Structure that is not immediately apparent, but when known helps explain the observed data
- *Latent* because its hidden
  - *"Latent:* (of a quality or state) existing but not yet developed or manifest; hidden; concealed."

Some other examples of latent structure

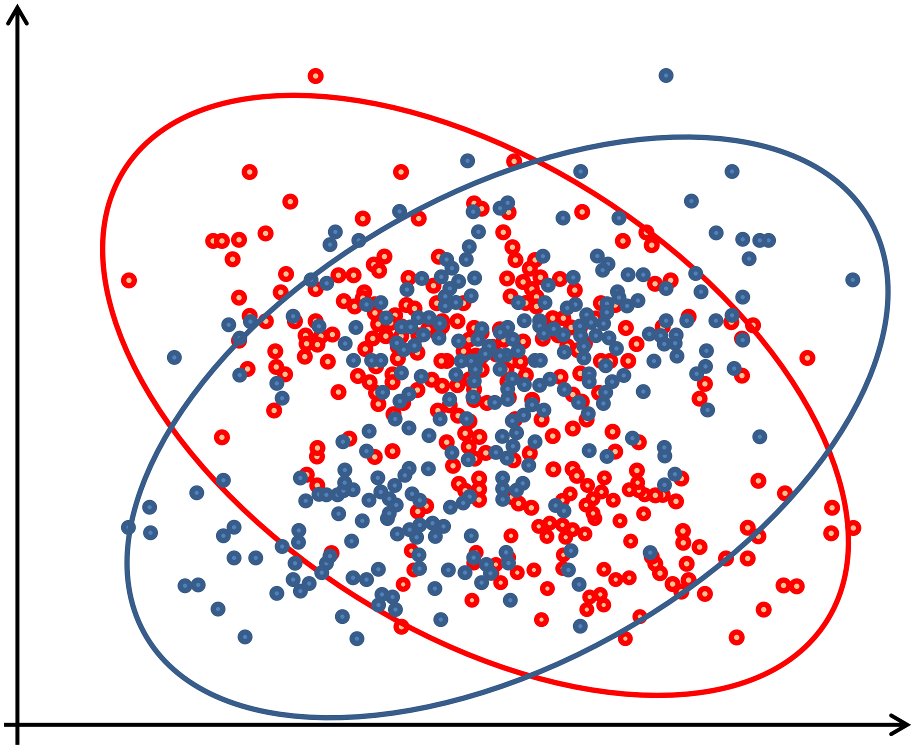
- Chiefly three underlying variables
  - Varying these can generate many images

Latent Structure in Distributions



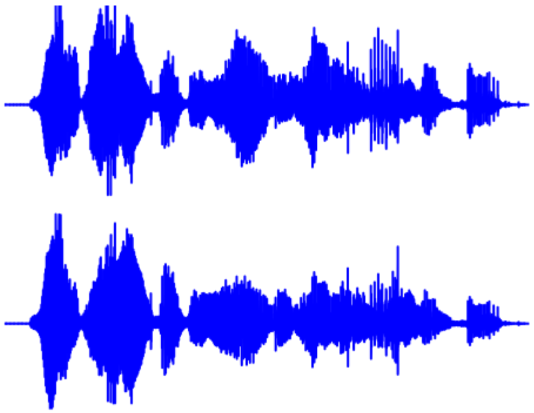
- A circular looking scatter of points

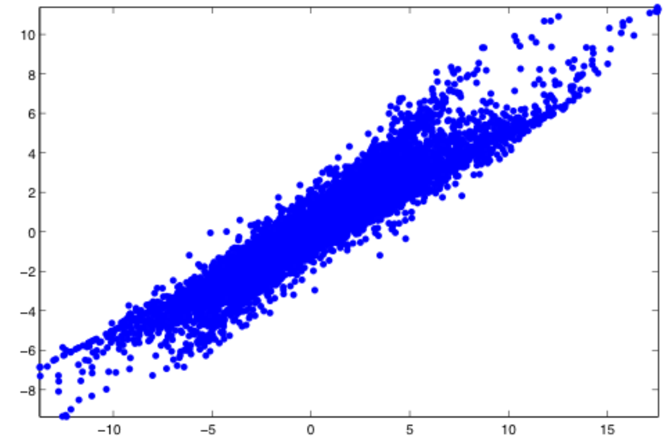
Latent Structure in Distributions



- The data are actually generated by two distributions
  - Generated under two different conditions
  - Knowledge of this helps one tease out factors

Latent Structure Explains Data





- The scatter of samples is better explained if we know there are two independent sources!

Latent Structure in Data

	Fed rate +	Fed rate -
Emerging markets +	++	+
Emerging market -	=	--

- Stock market table..
  - Knowing the typical effect of different factors on the stock market enables us to understand trends
    - And predict them
      - And make money
        - » Or lose it..

A Gaussian Variable

$$P(X) = N(X; aF_1 + bF_2 + c, \Theta)$$

- Several latent "factors" affecting the data
  - ***Factors are continuous variables***
  - E.g.   X = [BP, Pulse]
  - F_1 = time from exertion
  - F_2 = duration of exertion
  - Typically would be many more factors

What is a latent *structure*

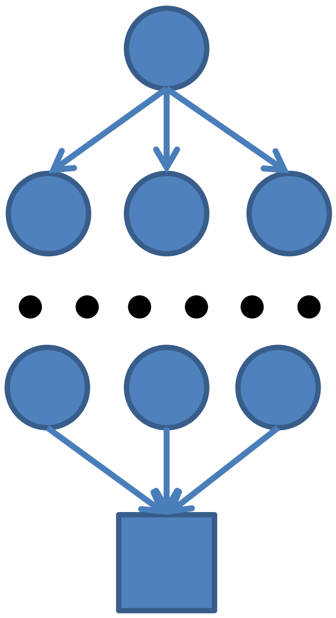
- Structure that is not observable, but can help explain data
  - Number of sources
  - Number of factors
  - Potentially observable
  - Could be hierarchical!

What is a Latent Variable Model?

- A *structured* model for observed data that assumes underlying *latent* structure
  - Latent structure expressed through *latent variables*
  - Generally affects observations by affecting parameters of a generating process

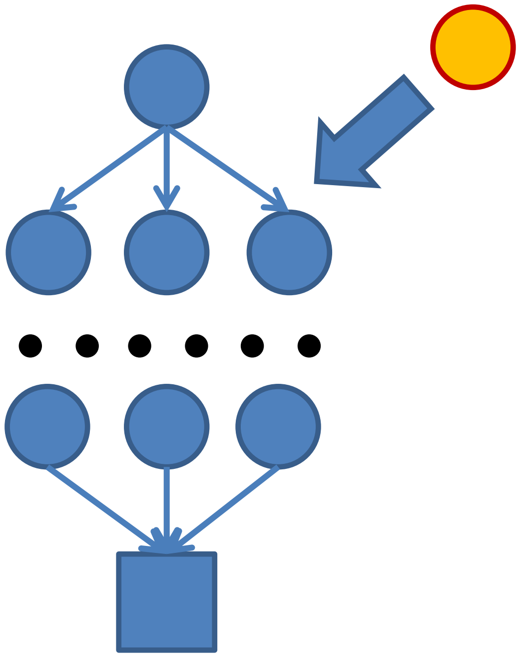
- The model structure may
  - Actually map onto *real* structure in the process
  - Impose structure artificially on the process to simplify the model
    - Make estimation/inference computationally tractable
    - “Simplify” → reduce the number of parameters

A Typical Symbolic Representation of a Latent Variable Model..



- Squares are observations, circles are latent variables

A Typical Symbolic Representation of a Latent Variable Model..



- Squares are observations, circles are latent variables
- Process may have inputs..

Latent Variables

- Latent variables may be categorical
  - E.g. “which books is being typed”
- Or continuous
  - E.g “time from exertion”

Examples of Extracting Latent Variables

- Principal Component Analysis / ICA
  - The “notes” are the latent factors
  - Knowing how many notes compose the music explains much of the data
- Factor Analysis
- Mixture models (mixture multinomials, mixture Gaussians, HMMs, hierarchical models, various “graphical” models)
- Techniques for estimation: Most commonly EM

Today

- A simple latent variable model applied to a very complex problem: Signal separation

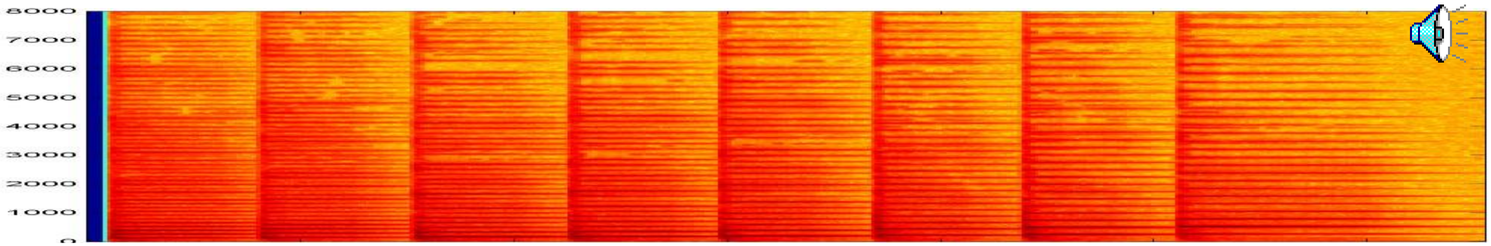
- With surprising success..

Sound separation and enhancement

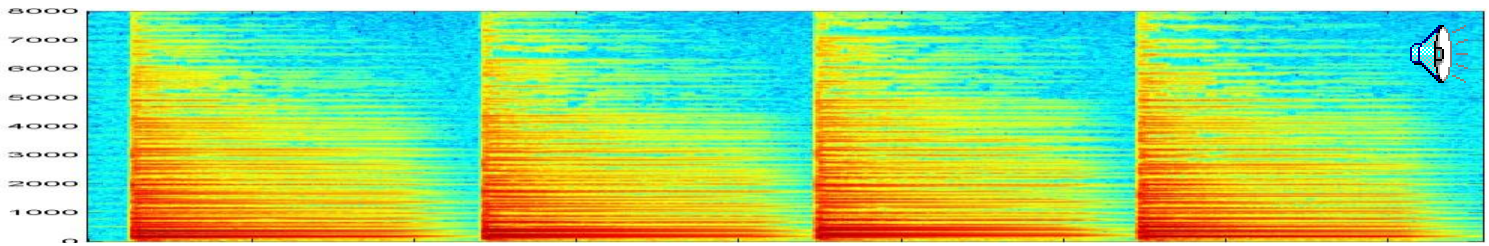
- A common problem: Separate or enhance sounds
  - Speech from noise
  - Suppress “bleed” in music recordings
  - Separate music components..
- Latent variable models: Do this with pots, pans, marbles and expectation maximization
  - *Probabilistic latent component analysis*
- Tools are applicable to other forms of data as well..

Sounds – an example

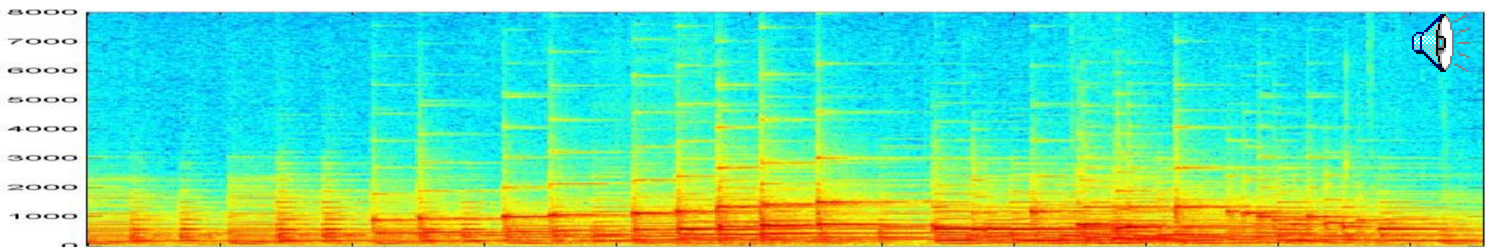
- A sequence of notes



- Chords from the same notes

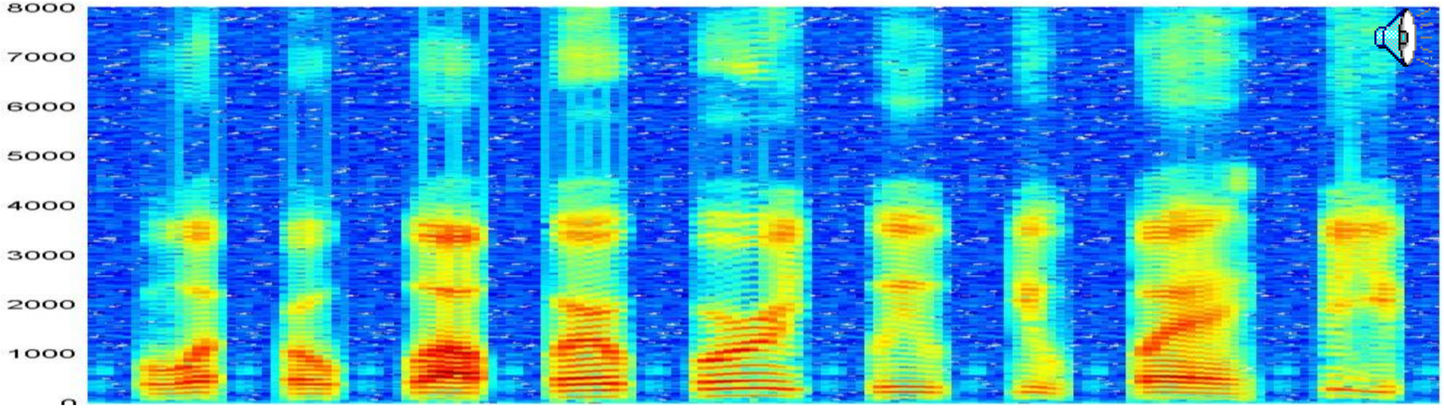


- A piece of music from the same (and a few additional) notes

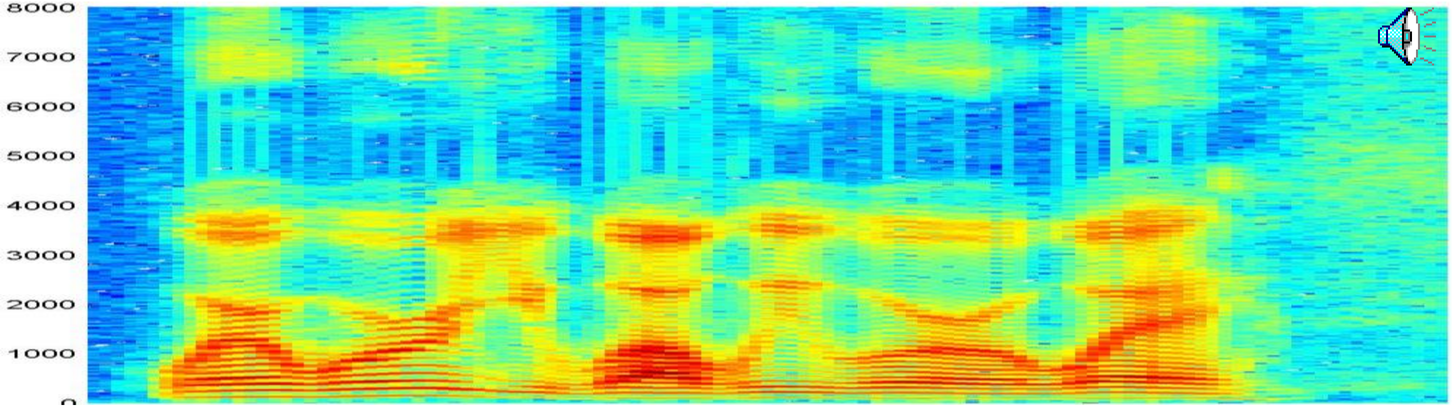


Sounds – an example

- A sequence of sounds



- A proper speech utterance from the same sounds



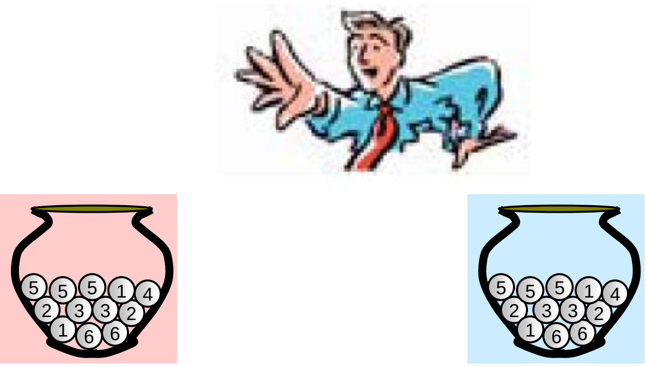
Template Sounds Combine to Form a Signal

- The individual component sounds "combine" to form the final complex sounds that we perceive
  - Notes form music
  - Phoneme-like structures combine in utterances

- Sound in general is composed of such "building blocks" or themes
  - Which can be simple – e.g. notes, or complex, e.g. phonemes
  - These units represent the ***latent building blocks of sounds***

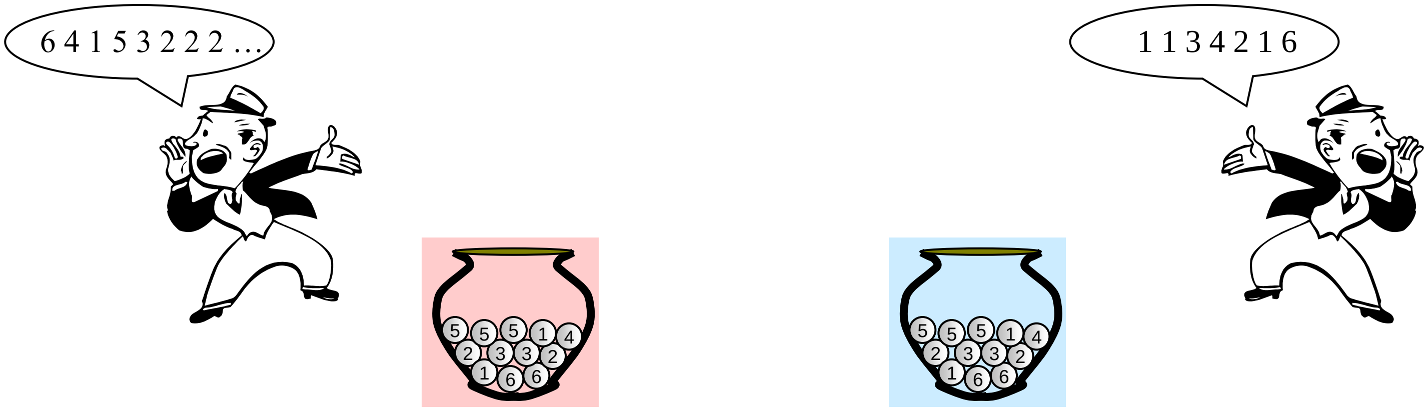
- Claim: Learning the building blocks enables us to manipulate sounds

The Mixture Multinomial



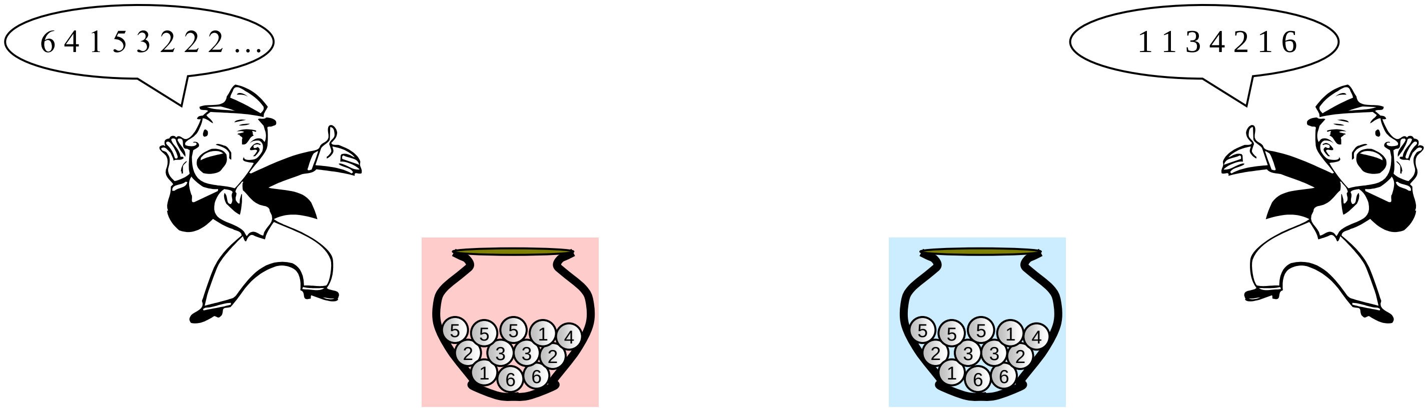
- A person drawing balls from a pair of urns
  - Each ball has a number marked on it
- You only hear the number drawn
  - No idea of which urn it came from
- Estimate various facets of this process..

More complex: TWO pickers



- Two *different* pickers are drawing balls from the *same* pots
  - After each draw they call out the number and replace the ball
- They select the pots with different probabilities
- From the numbers they call we must determine
  - Probabilities with which each of them select pots
  - The distribution of balls within the pots

Solution



- Analyze each of the callers separately
- Compute the probability of selecting pots separately for each caller
- But *combine* the counts of balls in the pots!!

Recap with only one picker and two pots

- Probability of Red urn:
  - P(1 | Red) = 1.71/7.31 = 0.234
  - P(2 | Red) = 0.56/7.31 = 0.077
  - P(3 | Red) = 0.66/7.31 = 0.090
  - P(4 | Red) = 1.32/7.31 = 0.181
  - P(5 | Red) = 0.66/7.31 = 0.090
  - P(6 | Red) = 2.40/7.31 = 0.328
- Probability of Blue urn:
  - P(1 | Blue) = 1.29/11.69 = 0.122
  - P(2 | Blue) = 0.56/11.69 = 0.322
  - P(3 | Blue) = 0.66/11.69 = 0.125
  - P(4 | Blue) = 1.32/11.69 = 0.250
  - P(5 | Blue) = 0.66/11.69 = 0.125
  - P(6 | Blue) = 2.40/11.69 = 0.056

- P(Z=Red) = 7.31/18 = 0.41
- P(Z=Blue) = 10.69/18 = 0.59

Called	P(red|X)	P(blue|X)
6	.8	.2
4	.33	.67
5	.33	.67
1	.57	.43
2	.14	.86
3	.33	.67
4	.33	.67
5	.33	.67
2	.14	.86
2	.14	.86
1	.57	.43
4	.33	.67
3	.33	.67
4	.33	.67
6	.8	.2
2	.14	.86
1	.57	.43
6	.8	.2

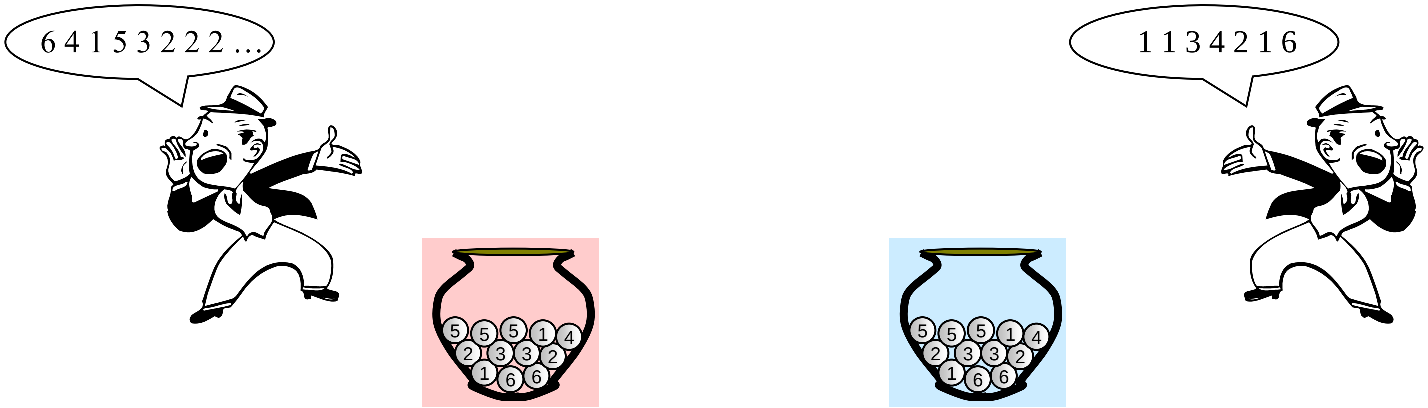
7.31 10.69

Two pickers

- Probability of drawing a number X for the first picker:
  – $P_1(X) = P_1(\text{red})P(X|\text{red}) + P_1(\text{blue})P(X|\text{blue})$
- Probability of drawing X for the second picker
  – $P_2(X) = P_2(\text{red})P(X|\text{red}) + P_2(\text{blue})P(X|\text{blue})$

- Note: $P(X|\text{red})$ and $P(X|\text{blue})$ are the same for both pickers
  – The pots are the same, and the probability of drawing a ball marked with a particular number is the same for both
- The probability of *selecting* a particular pot is different for both pickers
  – $P_1(X)$ and $P_2(X)$ are not related

Two pickers



- Probability of drawing a number X for the first picker:
  - $P_1(X) = P_1(red)P(X|red) + P_1(blue)P(X|blue)$
- Probability of drawing X for the second picker
  - $P_2(X) = P_2(red)P(X|red) + P_2(blue)P(X|blue)$

- Problem: From set of numbers called out by both pickers estimate
  - $P_1(color)$ and $P_2(color)$ for both colors
  - $P(X \mid red)$ and $P(X \mid blue)$ for all values of X

With TWO pickers

Called	P(red|X)	P(blue|X)
6	.8	.2
4	.33	.67
5	.33	.67
1	.57	.43
2	.14	.86
3	.33	.67
4	.33	.67
5	.33	.67
2	.14	.86
2	.14	.86
1	.57	.43
4	.33	.67
3	.33	.67
4	.33	.67
6	.8	.2
2	.14	.86
1	.57	.43
6	.8	.2

PICKER 1 7.31 10.69

PICKER 2

Called	P(red|X)	P(blue|X)
4	.57	.43
4	.57	.43
3	.57	.43
2	.27	.73
1	.75	.25
6	.90	.10
5	.57	.43

4.20 2.80

- Two tables
- The probability of *selecting* pots is independently computed for the two pickers

With TWO pickers

Called	P(red|X)	P(blue|X)
6	.8	.2
4	.33	.67
5	.33	.67
1	.57	.43
2	.14	.86
3	.33	.67
4	.33	.67
5	.33	.67
2	.14	.86
2	.14	.86
1	.57	.43
4	.33	.67
3	.33	.67
4	.33	.67
6	.8	.2
2	.14	.86
1	.57	.43
6	.8	.2

PICKER 1

7.31 10.69

PICKER 2

Called	P(red|X)	P(blue|X)
4	.57	.43
4	.57	.43
3	.57	.43
2	.27	.73
1	.75	.25
6	.90	.10
5	.57	.43

4.20 2.80

P(RED | PICKER1) = 7.31 / 18

P(BLUE | PICKER1) = 10.69 / 18

P(RED | PICKER2) = 4.2 / 7

P(BLUE | PICKER2) = 2.8 / 7

With TWO pickers

Called	P(red|X)	P(blue|X)
6	.8	.2
4	.33	.67
5	.33	.67
1	.57	.43
2	.14	.86
3	.33	.67
4	.33	.67
5	.33	.67
2	.14	.86
2	.14	.86
1	.57	.43
4	.33	.67
3	.33	.67
4	.33	.67
6	.8	.2
2	.14	.86
1	.57	.43
6	.8	.2

Called	P(red|X)	P(blue|X)
4	.57	.43
4	.57	.43
3	.57	.43
2	.27	.73
1	.75	.25
6	.90	.10
5	.57	.43

- To compute probabilities of numbers ***combine*** the tables
- Total count of Red: 11.51
- Total count of Blue: 13.49

With TWO pickers: The SECOND picker

Called	P(red|X)	P(blue|X)
6	.8	.2
4	.33	.67
5	.33	.67
1	.57	.43
2	.14	.86
3	.33	.67
4	.33	.67
5	.33	.67
2	.14	.86
2	.14	.86
1	.57	.43
4	.33	.67
3	.33	.67
4	.33	.67
6	.8	.2
2	.14	.86
1	.57	.43
6	.8	.2

Called	P(red|X)	P(blue|X)
4	.57	.43
4	.57	.43
3	.57	.43
2	.27	.73
1	.75	.25
6	.90	.10
5	.57	.43

- Total count for "Red" : 11.51
- Red:
  - Total count for 1: 2.46
  - Total count for 2: 0.83
  - Total count for 3: 1.23
  - Total count for 4: 2.46
  - Total count for 5: 1.23
  - Total count for 6: 3.30
  - P(6|RED) = 3.3 / 11.51 = 0.29

In Squiggles

- Given a sequence of observations $O_{k,1}$, $O_{k,2}$, .. from the k^{th} picker
  - $N_{k,X}$ is the number of observations of color X drawn by the k^{th} picker
- Initialize $P_k(Z)$, $P(X|Z)$ for pots Z and colors X
- Iterate:
  - For each Color X, for each pot Z and each observer k:

$$P_k(Z \mid X) = \frac{P(X \mid Z) P_k(Z)}{\sum_{Z'} P_k(Z') P(X \mid Z')}$$

  - Update probability of numbers for the pots:

$$P(X \mid Z) = \frac{\sum_{k} N_{k,X} P_k(Z \mid X)}{\sum_{k} \sum_{Z'} N_{k,X} P_k(Z' \mid X)}$$

  - Update the mixture weights: probability of urn selection for each picker

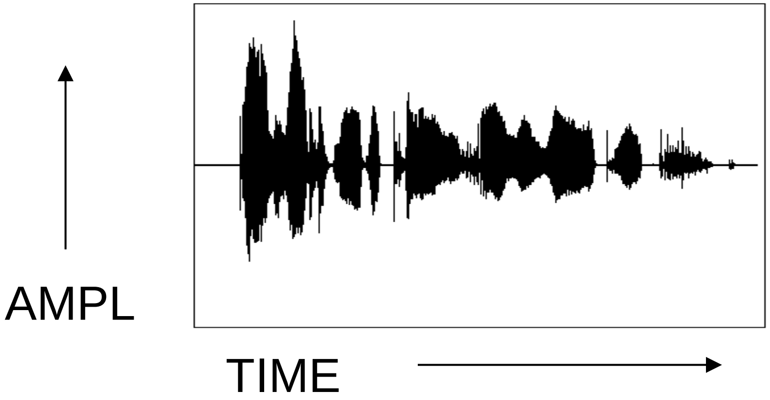
$$P_k(Z) = \frac{\sum_{X} N_{k,X} P_k(Z \mid X)}{\sum_{Z'} \sum_{X} N_{k,X} P_k(Z' \mid X)}$$

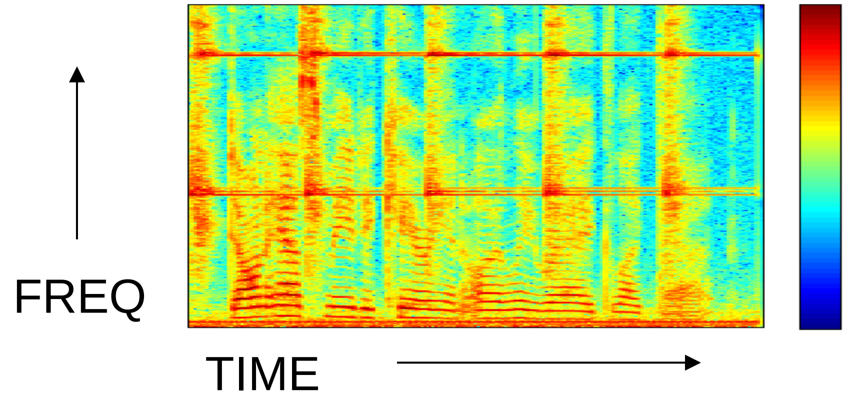
Signal Separation with the Urn model

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

- We shall see..

The representation



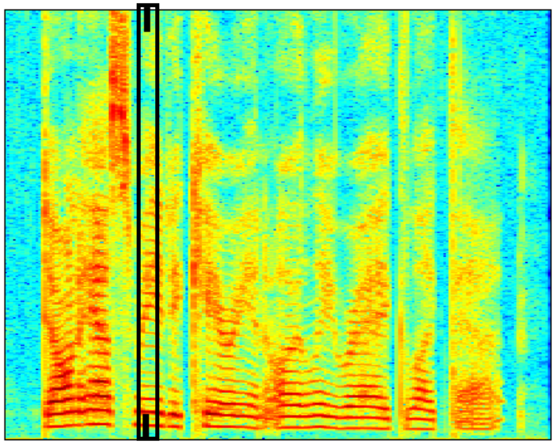


- We represent signals spectrographically
  - Sequence of magnitude spectral vectors estimated from (overlapping) segments of signal
  - Computed using the short-time Fourier transform
  - Note: Only retaining the magnitude of the STFT for operations
  - We will, need the phase later for conversion to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

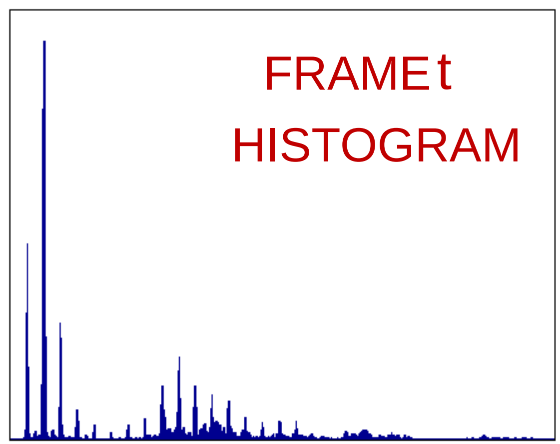
A Multinomial Model for Spectra

- A generative model for one frame of a spectrogram
  - 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

f



FRAME t



f

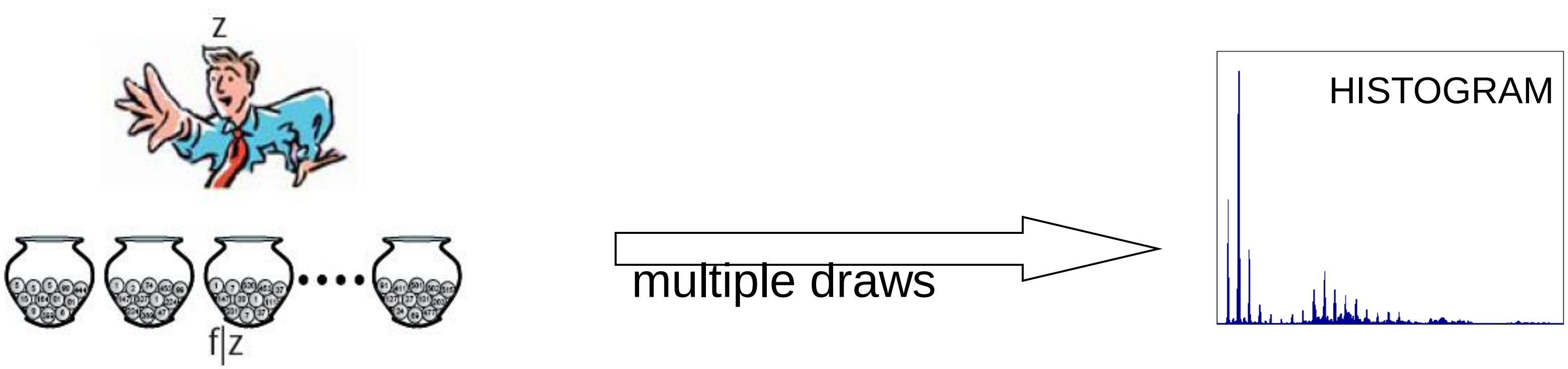
$P_t(f)$

The balls are marked with discrete frequency indices from the DFT

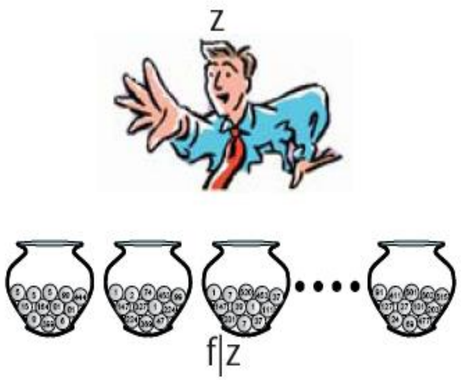
Probability distribution underlying the t-th spectral vector

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

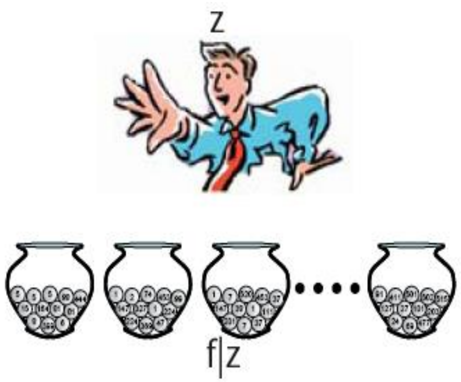


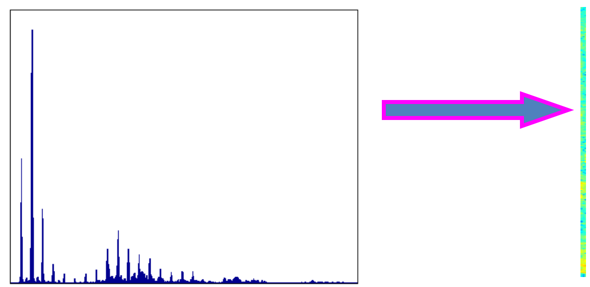
The Picker Generates a Spectrogram



- 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

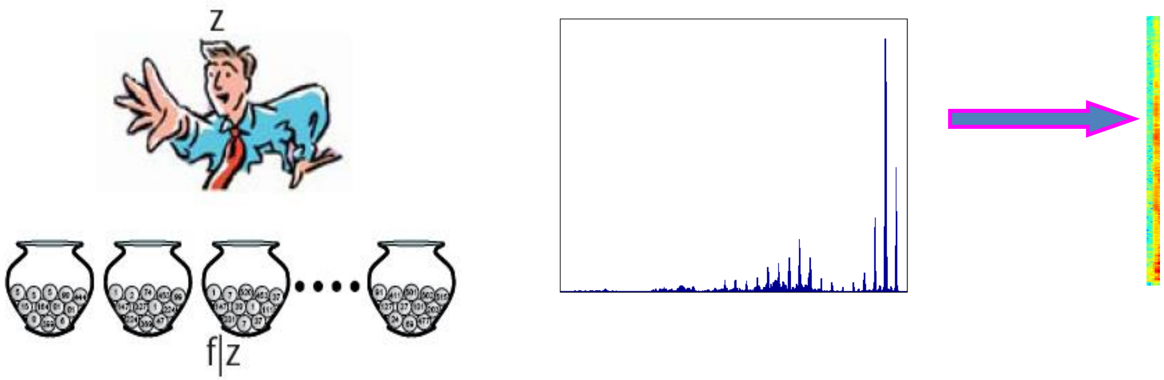
The Picker Generates a Spectrogram





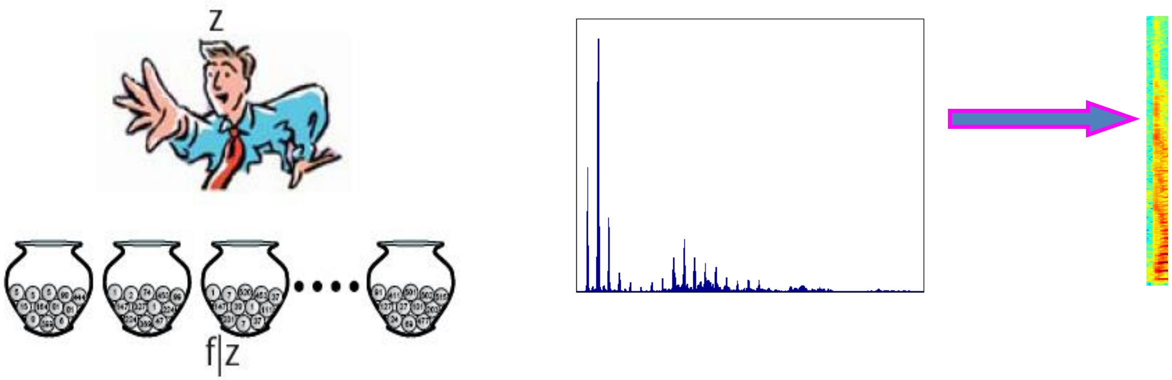
- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to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

The Picker Generates a Spectrogram



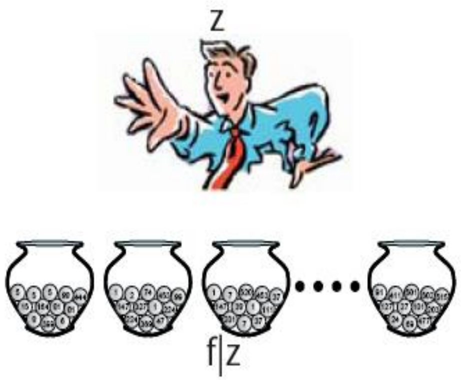
- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to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

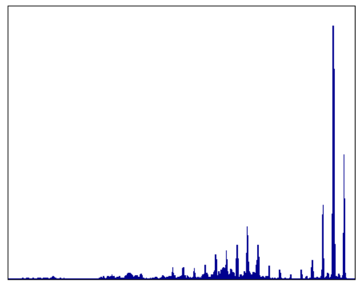
The Picker Generates a Spectrogram

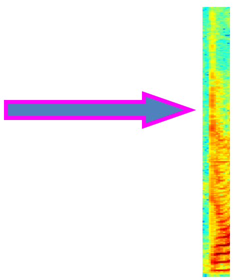


- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to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

The Picker Generates a Spectrogram

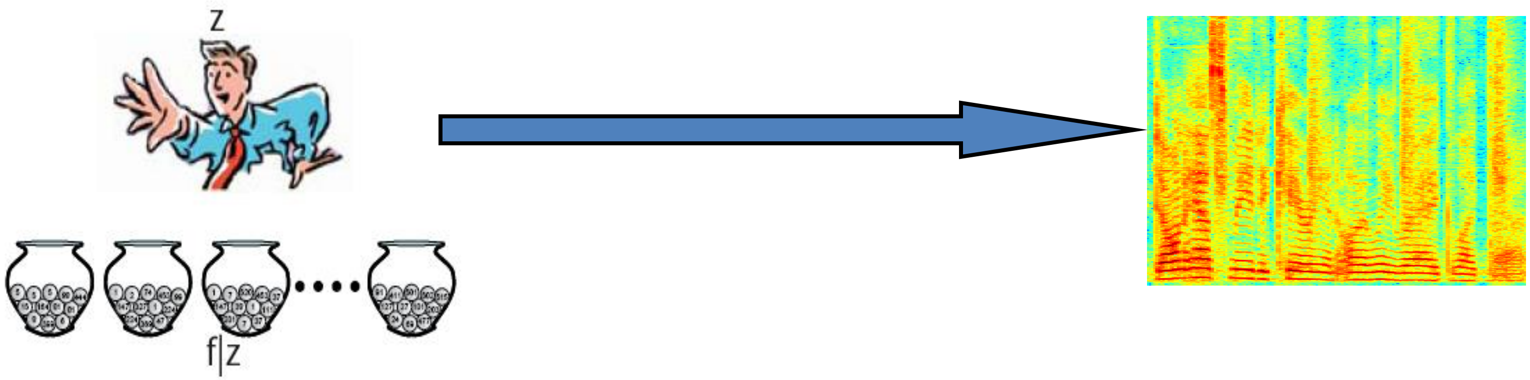






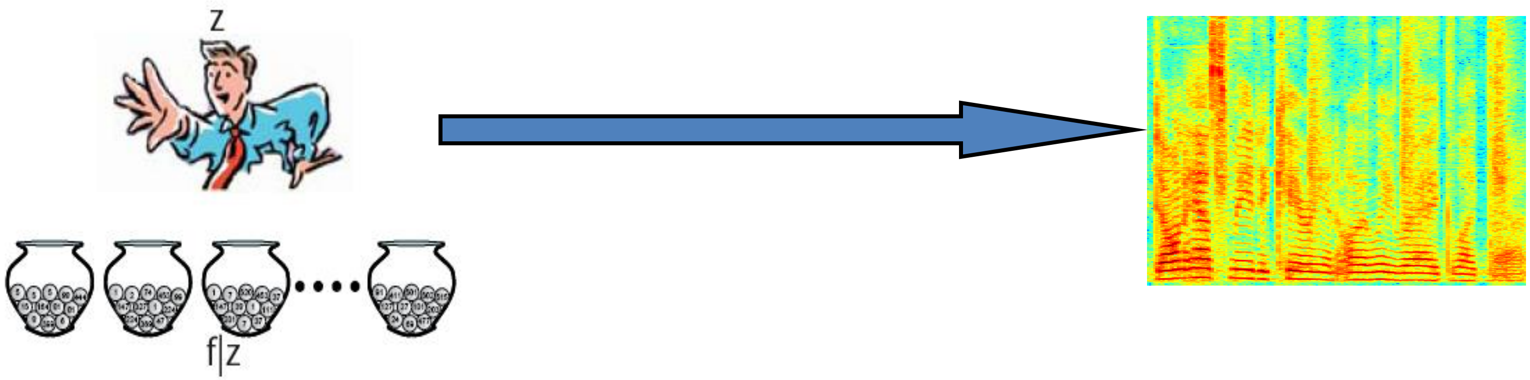
- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to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

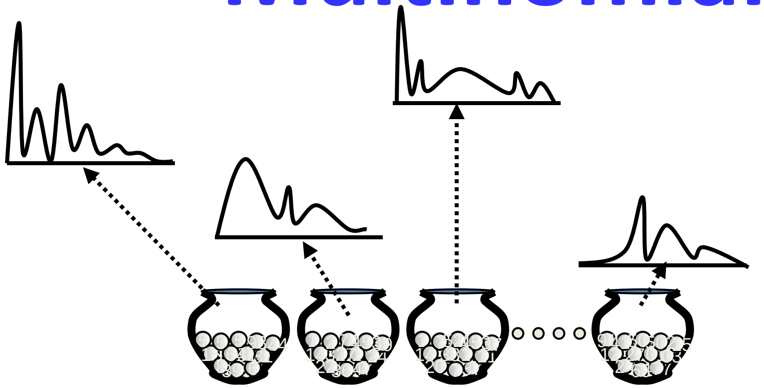
$$P_t(f) = \sum_z P_t(z) P(f \mid z)$$

Frame-specific spectral distribution

SOURCE specific bases

Frame(time) specific mixture weight

Spectral View of *Component* Multinomials

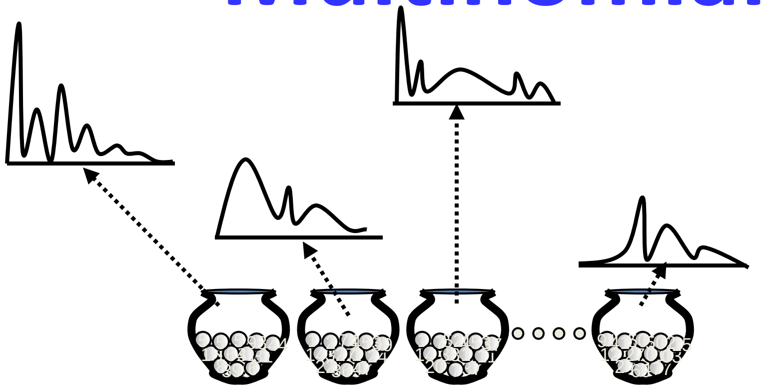


- Each component multinomial (urn) is actually a normalized histogram over frequencies $P(f \mid z)$
  - I.e. a spectrum

- 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

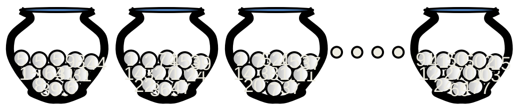


- By "learning" the mixture multinomial model for any sound source we "discover" these latent spectral structures for the source

- 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 $P_t(z)$

EM Update Equations

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

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

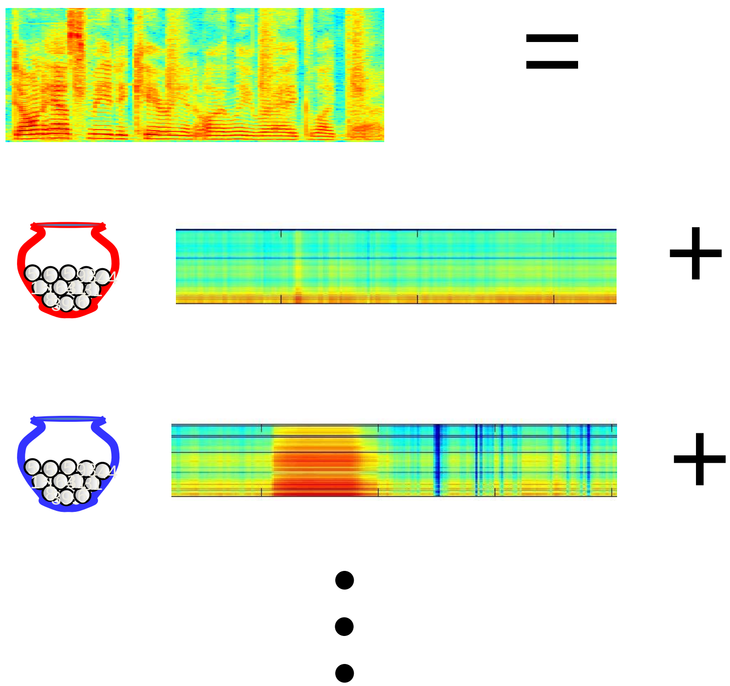
  - Compute mixture weight of z^{th} urn

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

  - Compute the probabilities of the frequencies for the z^{th} urn

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

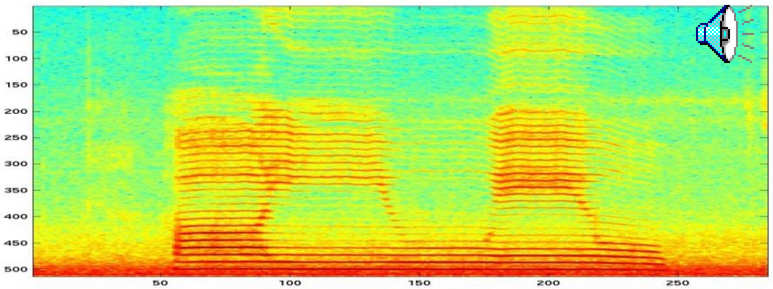
How the bases compose the signal

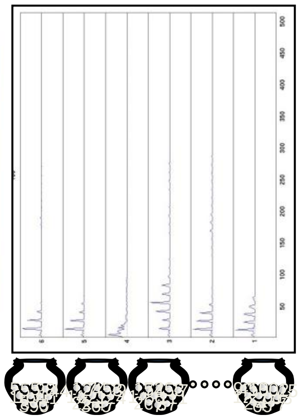


- The overall signal is the sum of the contributions of individual urns
  - Each urn contributes a different amount to each frame
- The contribution of the z-th urn to the t-th frame is given by $P(f|z)P_t(z)S_t$
  - $S_t = \Sigma_f S_t\,(f)$

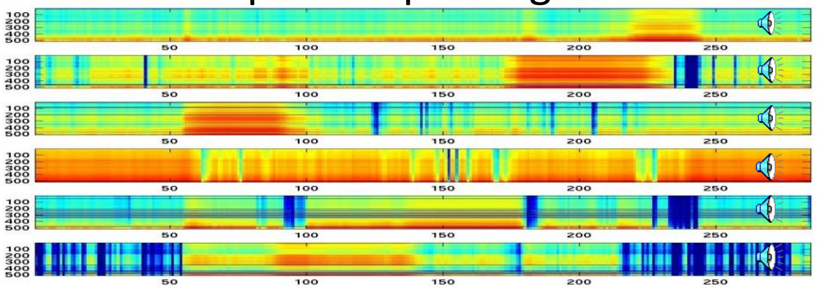
Learning Structures

Speech Signal

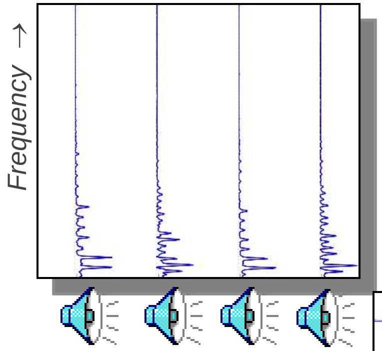




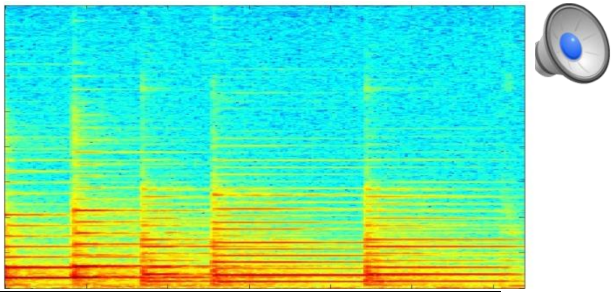
Basis-specific spectrograms

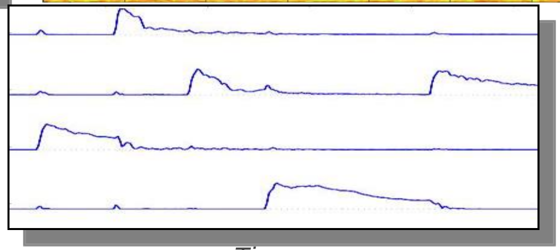


P(f|z)



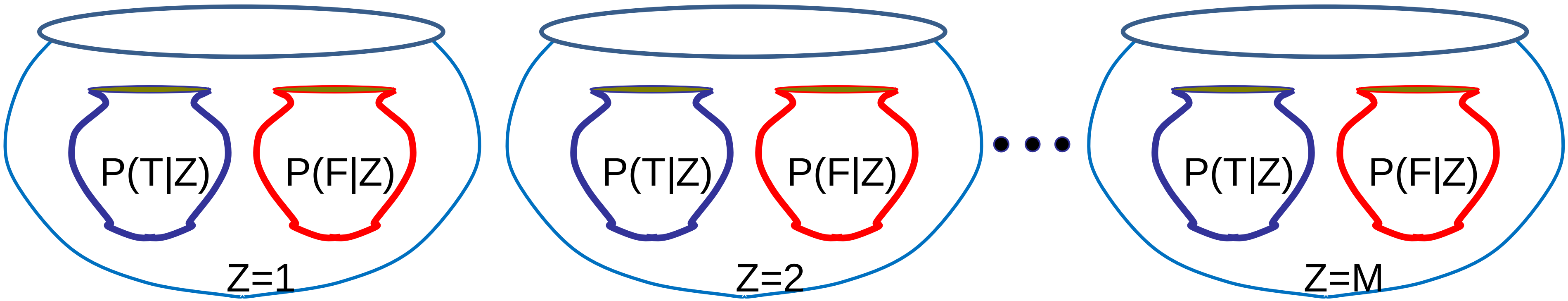
From Bach's Fugue in Gm



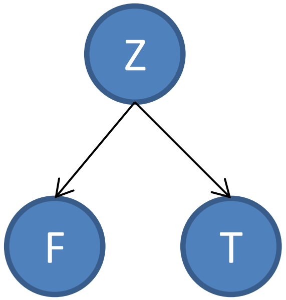


$P_t(z)$

Bag of Spectrograms PLCA Model

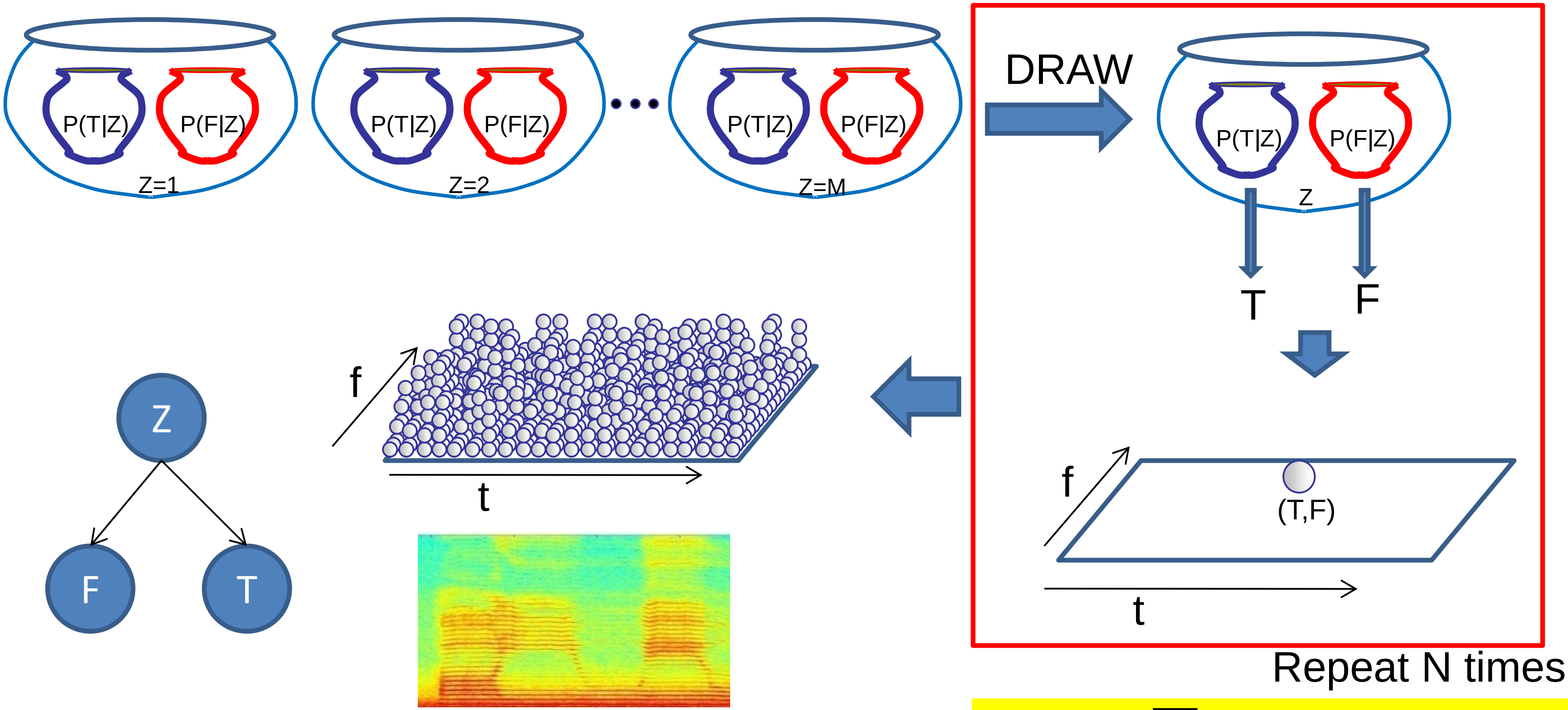


- Compose the entire spectrogram all at once
- Urns include two types of balls
  - One set of balls represents frequency F
  - The second has a distribution over time T
- Each draw:
  - Select an urn
  - Draw "F" from frequency pot
  - Draw "T" from time pot
  - Increment histogram at (T,F)



$$P(t, f) = \sum_{Z} P(z)P(t \mid z)P(f \mid z)$$

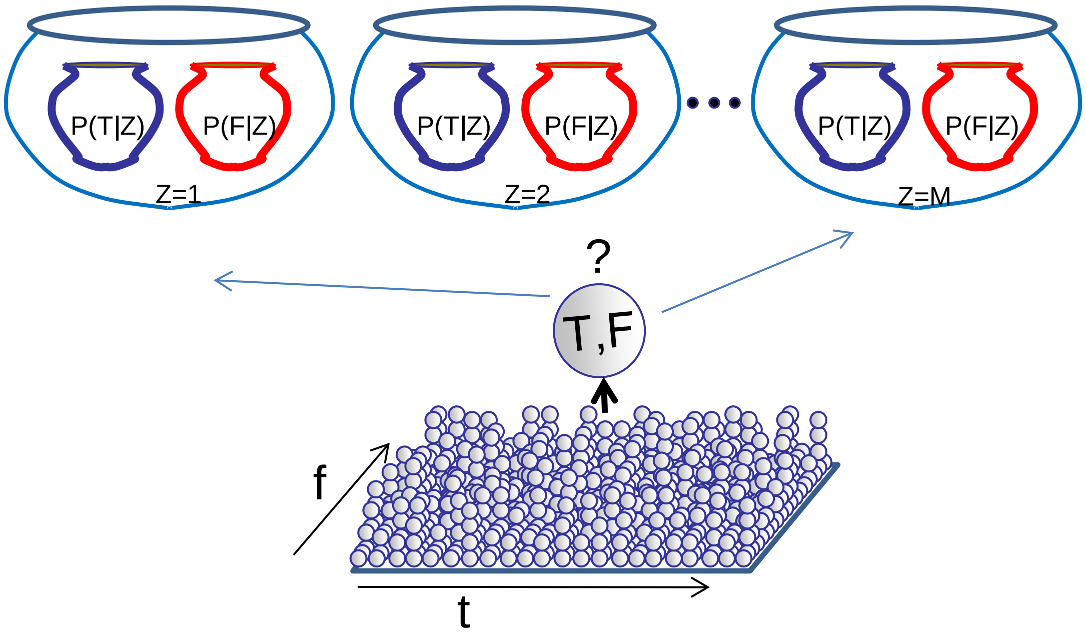
The bag of spectrograms



$$P(t,f) = \sum_{Z} P(z)P(t|z)P(f|z)$$

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

Estimating the bag of spectrograms



$$P(t,f)=\sum_{Z}P(z)P(t\,|\,z)P(f\,|\,z)$$

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

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

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

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

- 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)

How meaningful are these structures

- Are these really the “notes” of sound

- **To investigate, lets go back in time..**

The Engineer and the Musician

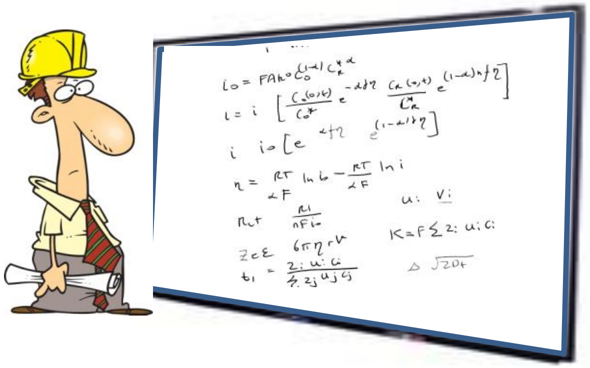
Once upon a time a rich potentate discovered a previously unknown recording of a beautiful piece of music. Unfortunately it was badly damaged.

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?

The Engineer and the Musician

- The Engineer works on the signal
  - ***Restore*** it
- The musician works on his familiarity with music
  - He knows how music is composed
  - He can identify notes and their cadence
    - But took many many years to learn these skills
  - He uses these skills to ***recompose*** the music

What the musician can do

- Notes are distinctive
- The musician knows notes (of all instruments)
- He can
  - *Detect notes in the recording*
    - Even if it is scratchy
    - Reconstruct damaged music
  - *Transcribe individual components*
    - Reconstruct separate portions of the music

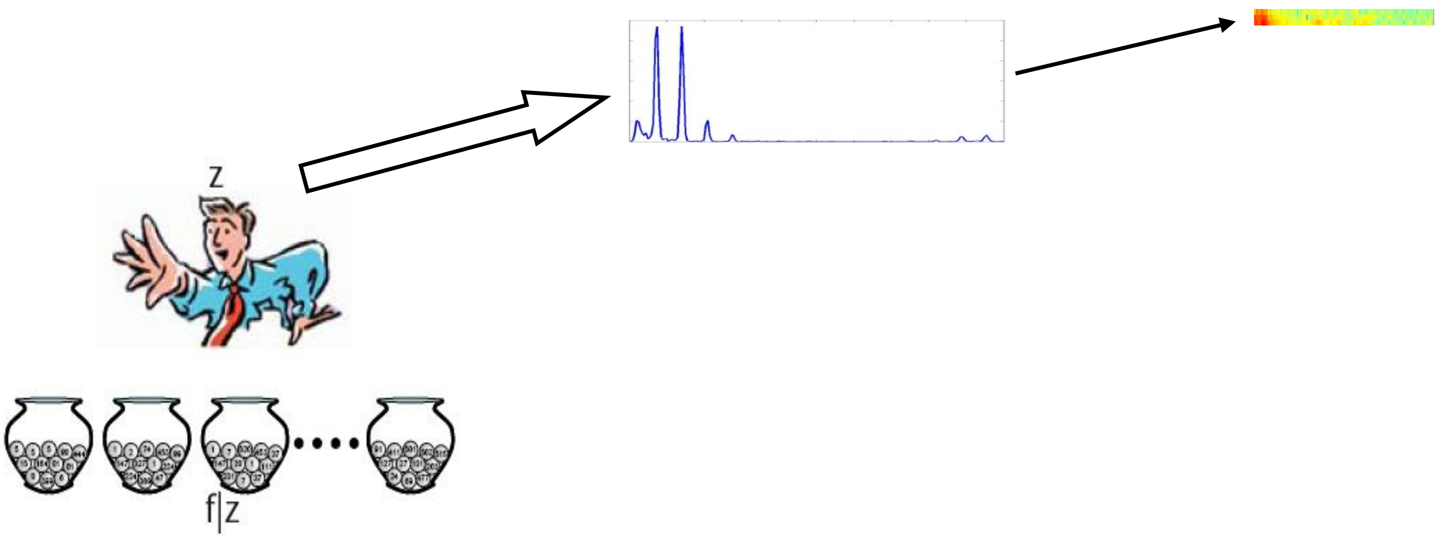
Music over a telephone

- The King actually got music over a telephone
- The musician must restore it..

- **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

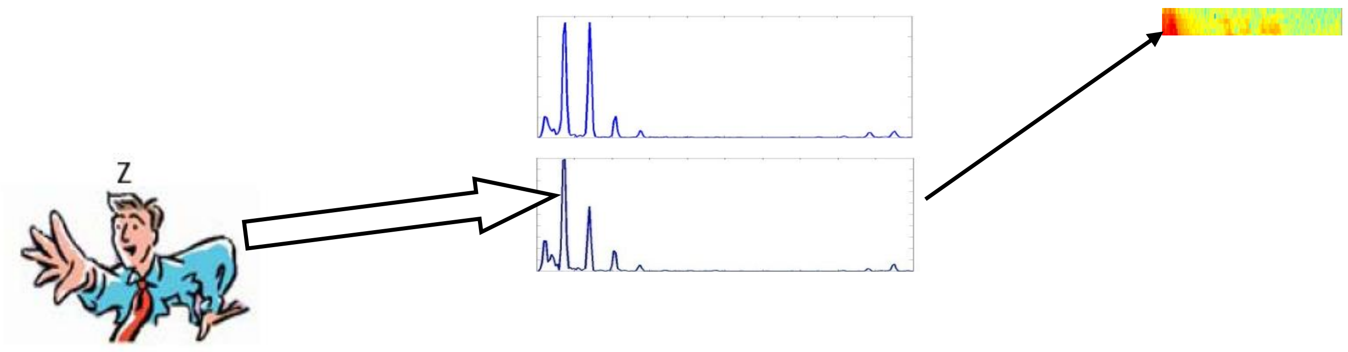
Bandwidth Expansion

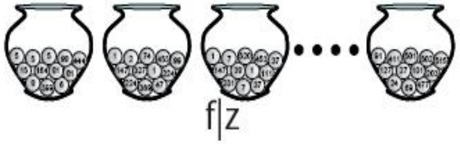
- The picker has drawn the histograms for every frame in the signal



Bandwidth Expansion

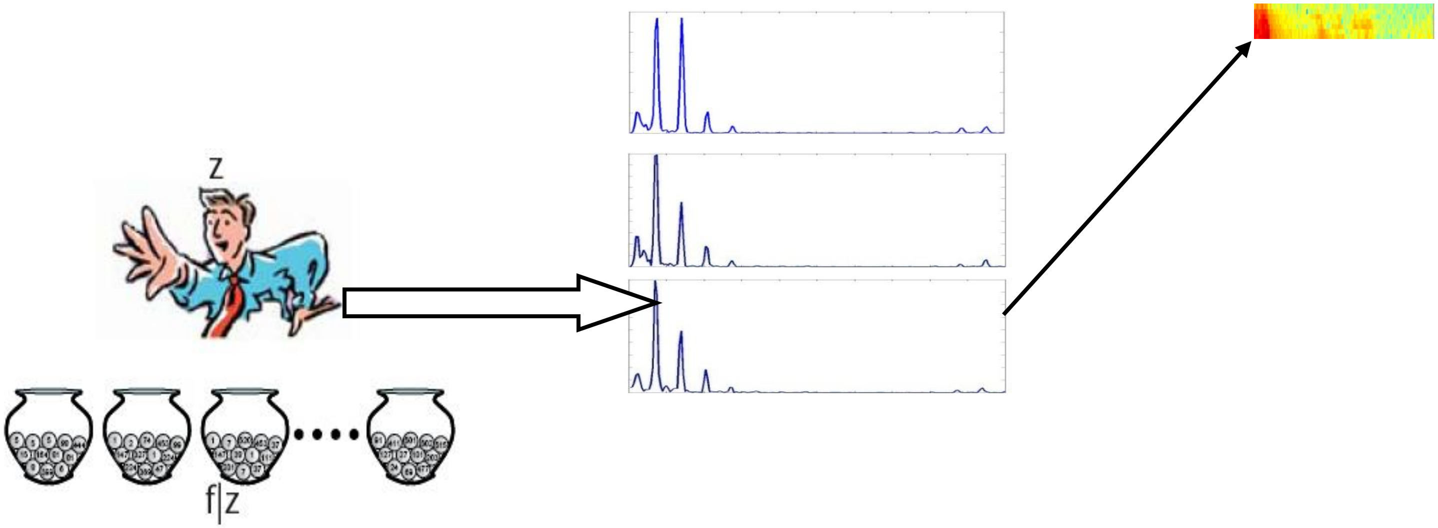
- The picker has drawn the histograms for every frame in the signal





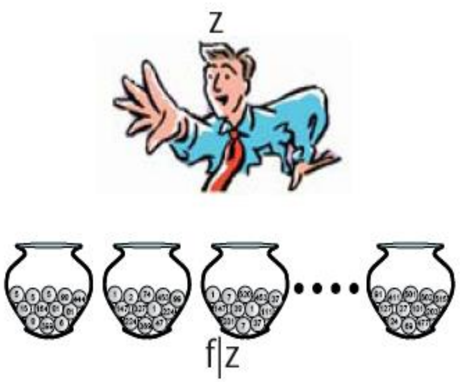
Bandwidth Expansion

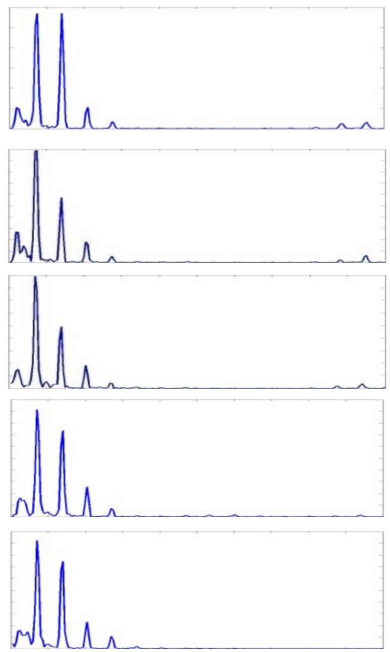
- The picker has drawn the histograms for every frame in the signal

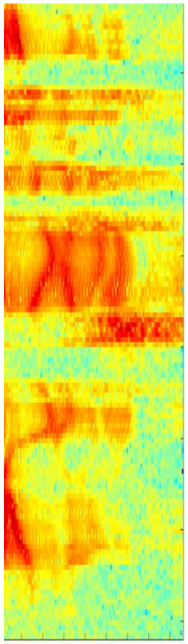


Bandwidth Expansion

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

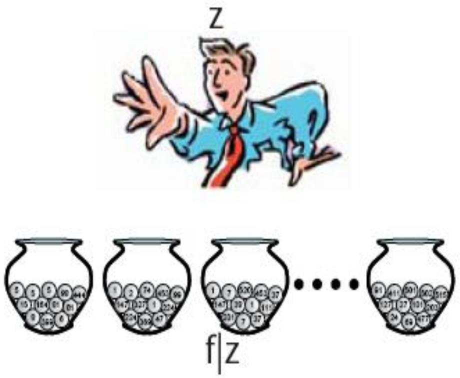


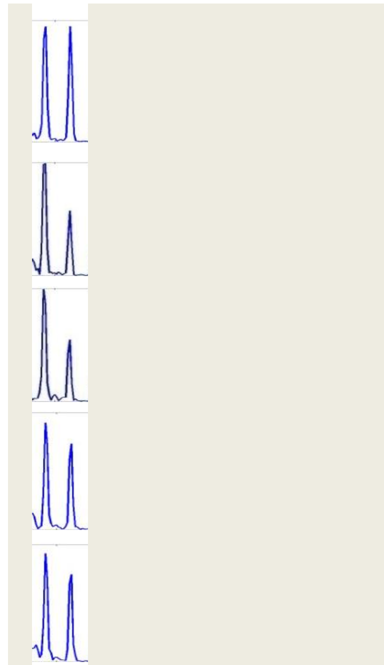


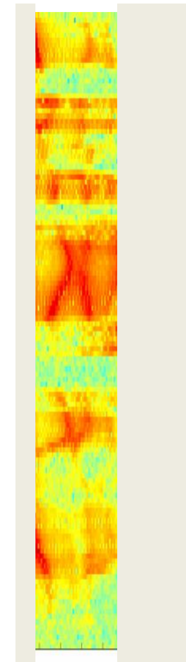


Bandwidth Expansion

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

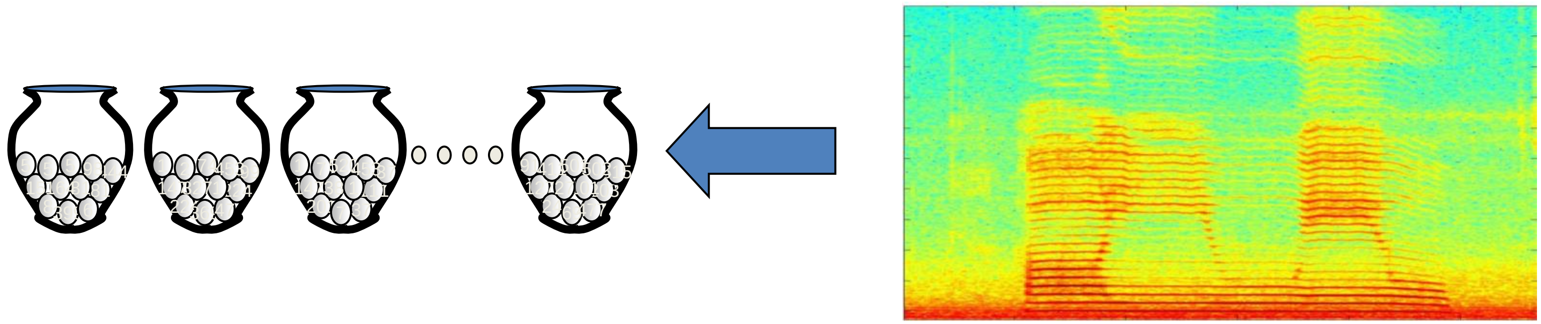






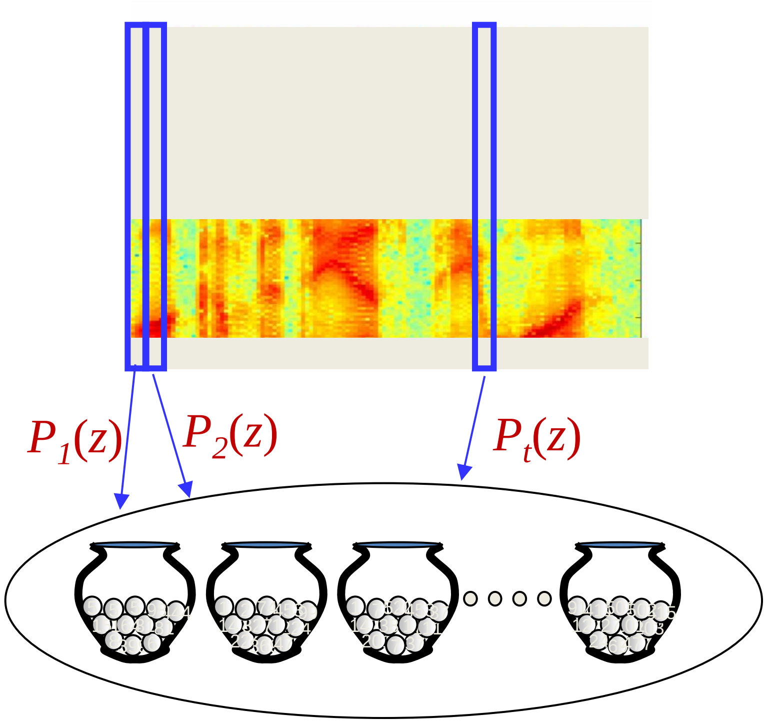
- However, we are only able to observe the number of draws of some frequencies and not the others
- We must estimate the draws of the unseen frequencies

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
    - Each magnitude spectral vector is a mixture of a common set of bases
    - Use the EM to learn bases from them
  - Basically learning the "notes"

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
  - Find out which notes were active at what time

Step 2

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

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

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

$$P_t(z) = \frac{\sum_{f \in (\text{observedfrequencies})} P_t(z \mid f)S_t(f)}{\sum_{z'} \sum_{f \in (\text{observedfrequencies})} P_t(z' \mid f)S_t(f)}$$

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

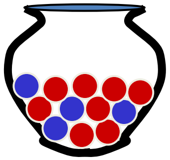
Step 3 and Step 4: Recompose

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

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

- 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 negative multinomial

- Given P(X) for all outcomes X
- Observed $n(X_1)$, $n(X_2)$..$n(X_k)$
- What is $n(X_{k+1})$, $n(X_{k+2})$...

$$P(n(X_{k+1}), n(X_{k+2}), \ldots) = \frac{\Gamma\left(N_o + \sum_{i>k} n(X_i)\right)}{\Gamma(N_o)\Gamma\left(\sum_{i>k} n(X_i)\right)} P_o \prod_{i>k} P(X_i)^{n(X_i)}$$

- N_o is the total number of observed counts
  - $n(X_1) + n(X_2) + \ldots$
- P_o is the total probability of observed events
  - $P(X_1) + P(X_2) + \ldots$

Estimating unobserved frequencies

- Expected value of the number of draws from a negative multinomial:

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

- Estimated spectrum in unobserved frequencies

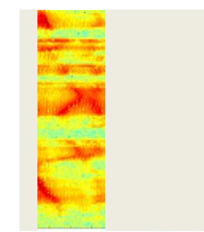
$$\hat{S}_t(f) = N_t P_t(f)$$

Overall Solution

- Learn the “urns” for the signal source from broadband training data

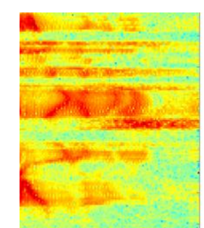
- For each frame of the reduced bandwidth test utterance, find mixture weights for the urns
  - Ignore (marginalize) the unseen frequencies

$P_t(z)$

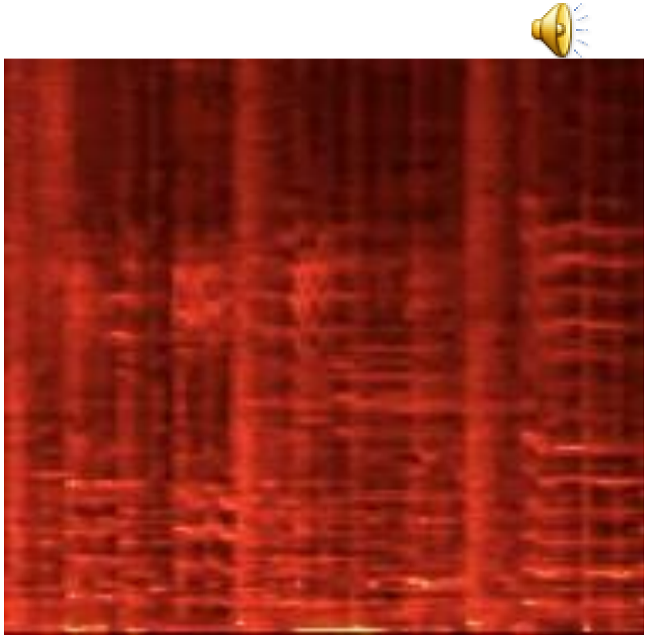


- Given the complete mixture multinomial distribution for each frame, estimate spectrum (histogram) at unseen frequencies

$P_t(z)$



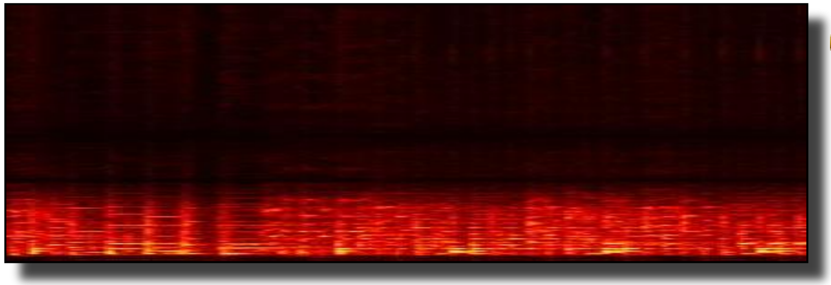
Prediction of Audio



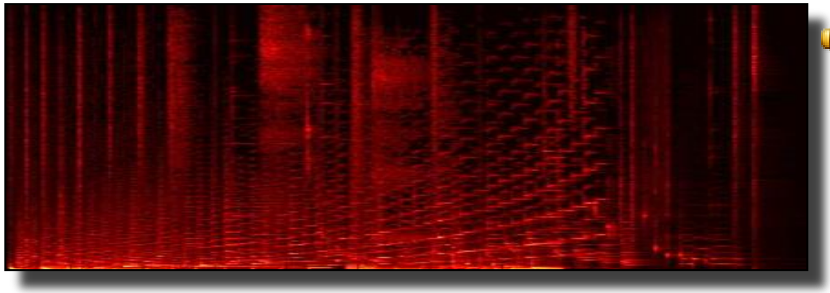
- An example with random spectral holes

Predicting frequencies

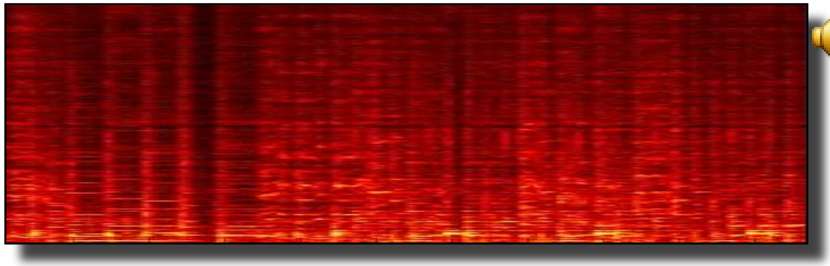
- Reduced BW data



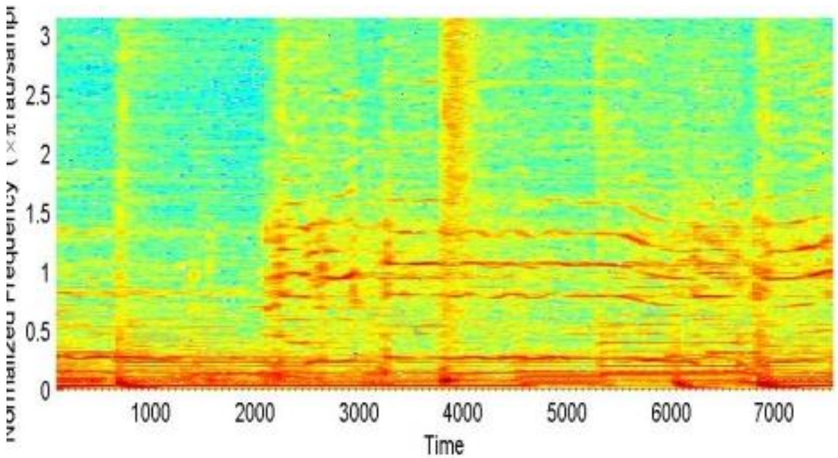
- Bases learned from this



- Bandwidth expanded version



Resolving the components

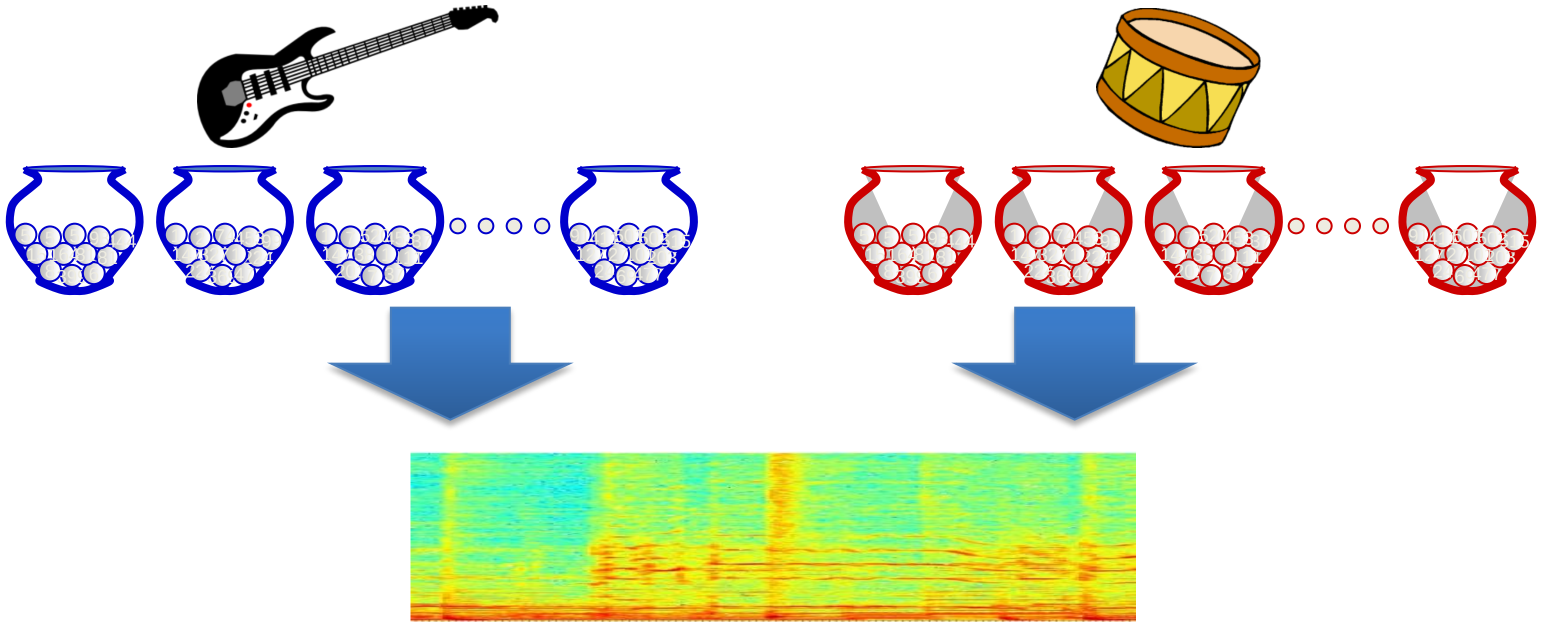


- The musician wants to follow the individual tracks in the recording..
  - Effectively "separate" or "enhance" them against the background

Signal Separation from Monaural Recordings

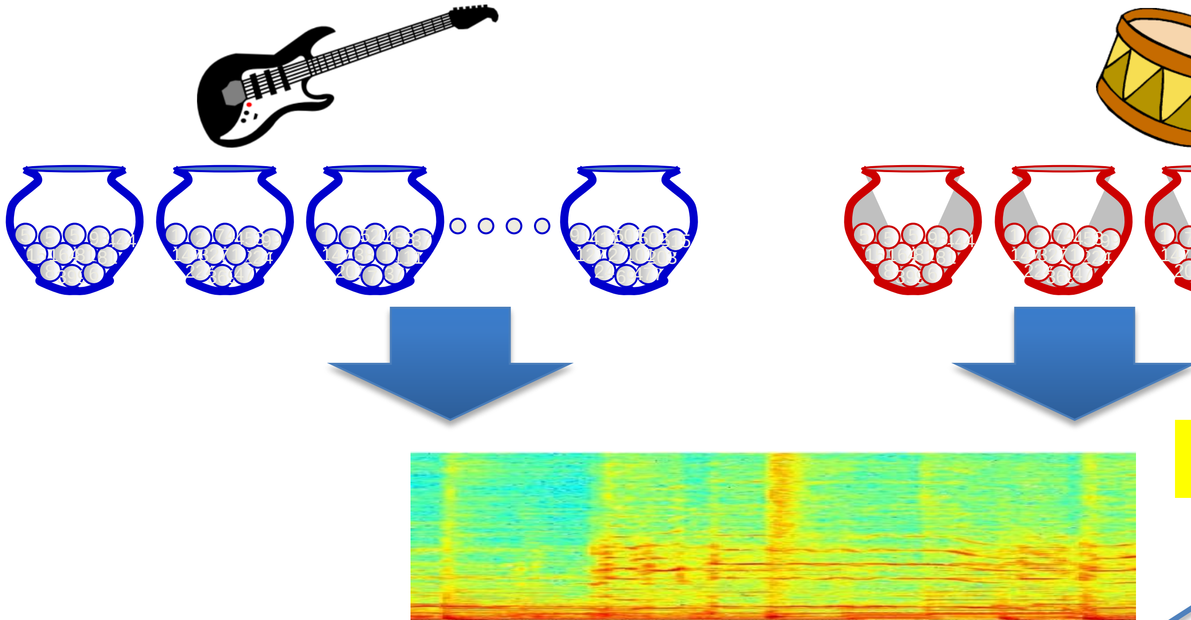
- Multiple sources are producing sound simultaneously
- The combined signals are recorded over a single microphone
- The goal is to selectively separate out the signal for a target source in the mixture
  – Or at least to enhance the signals from a selected source

Supervised separation: Example with two sources



- Each source has its own bases
  - **Can be learned from unmixed recordings of the source**
- All bases combine to generate the mixed signal
- Goal: Estimate the contribution of individual sources

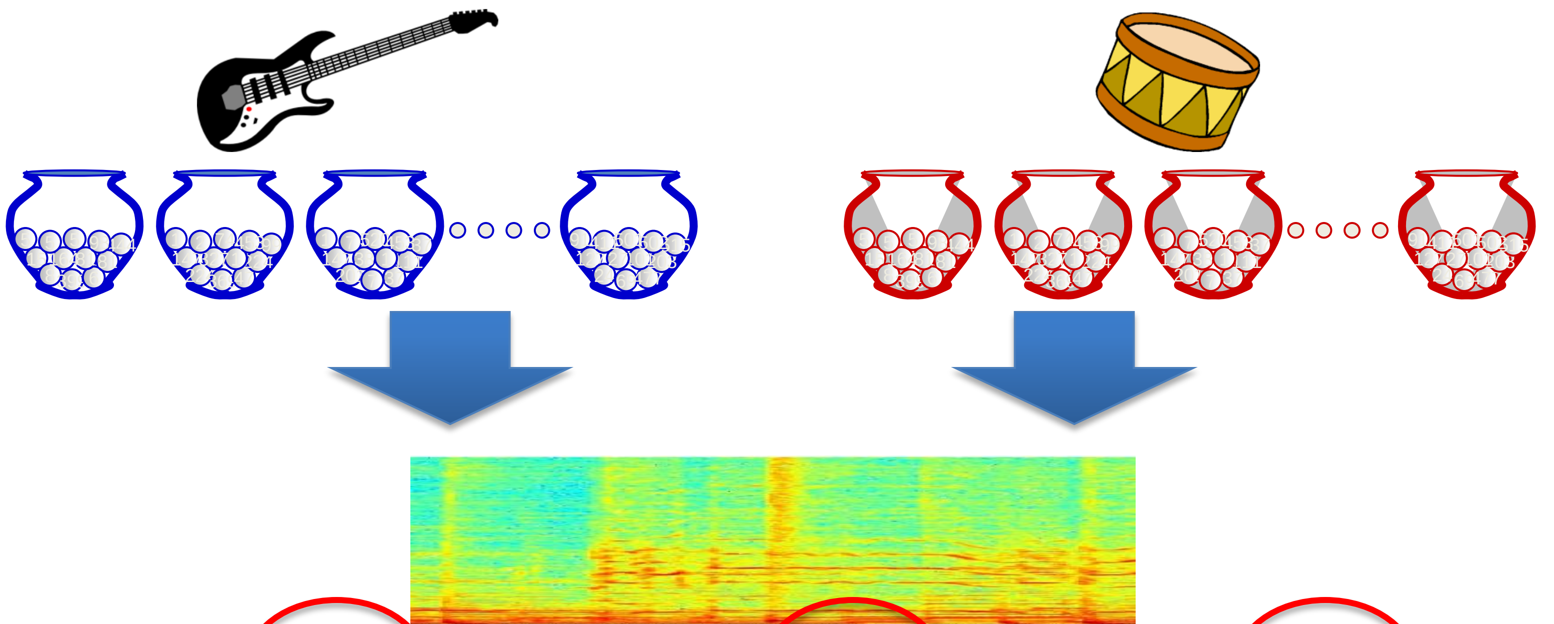
Supervised separation: Example with two sources



KNOWN A PRIORI

$$P_t(f) = \sum_{all\,z} P_t(z)P(f\,|\,z) = \sum_{z\ for\ source1} P_t(z)P(f\,|\,z) + \sum_{z\ for\ source2} P_t(z)P(f\,|\,z)$$

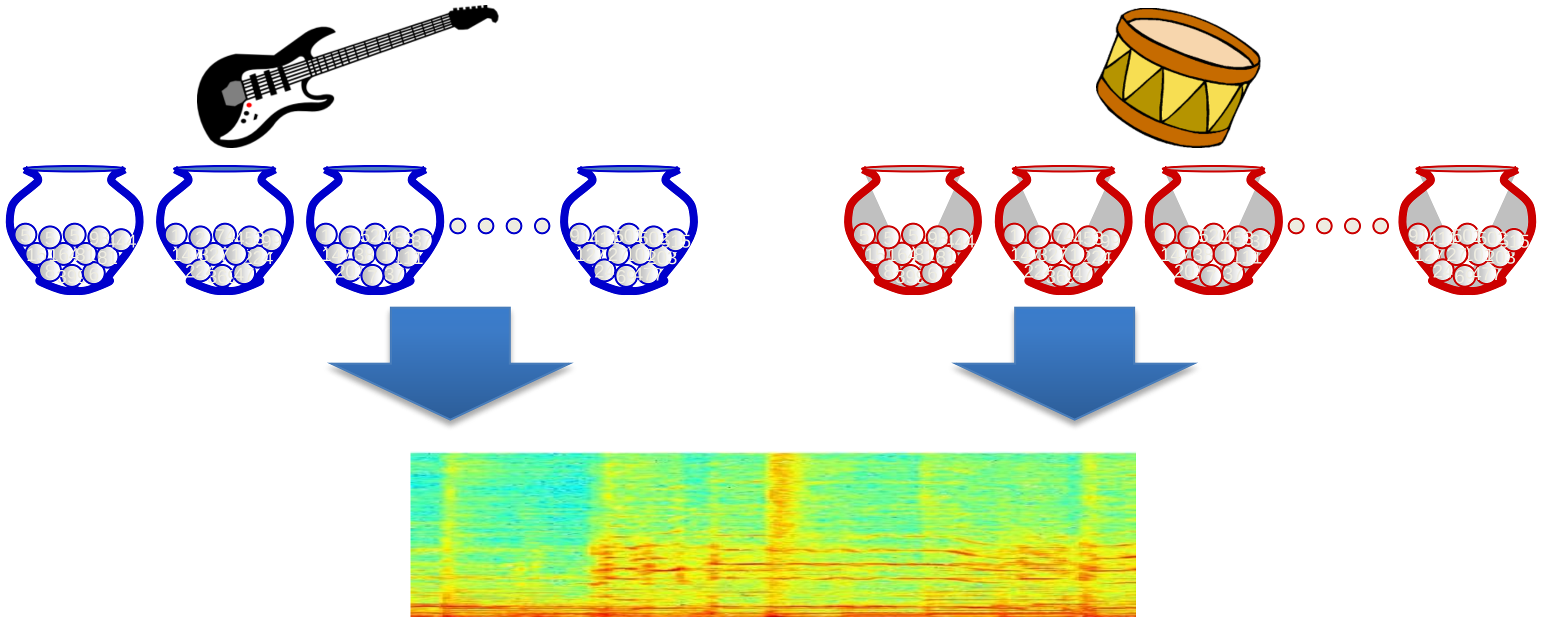
Supervised separation: Example with two sources



$$P_t(f) = \sum_{all\, z} P_t(z) P(f \mid z) = \sum_{z\ for\ source1} P_t(z) P(f \mid z) + \sum_{z\ for\ source2} P_t(z) P(f \mid z)$$

- Find mixture weights for all bases for each frame

Supervised separation: Example with two sources



$$P_t(f) = \sum_{all\ z} P_t(z)P(f\,|\,z) = \sum_{z\ for\ source1} P_t(z)P(f\,|\,z) + \sum_{z\ for\ source2} P_t(z)P(f\,|\,z)$$

- Find mixture weights for all bases for each frame
- Segregate contribution of bases from each source

$$\boxed{P_t^{source1}(f) = \sum_{z\ for\ source1} P_t(z)P(f\,|\,z)}$$

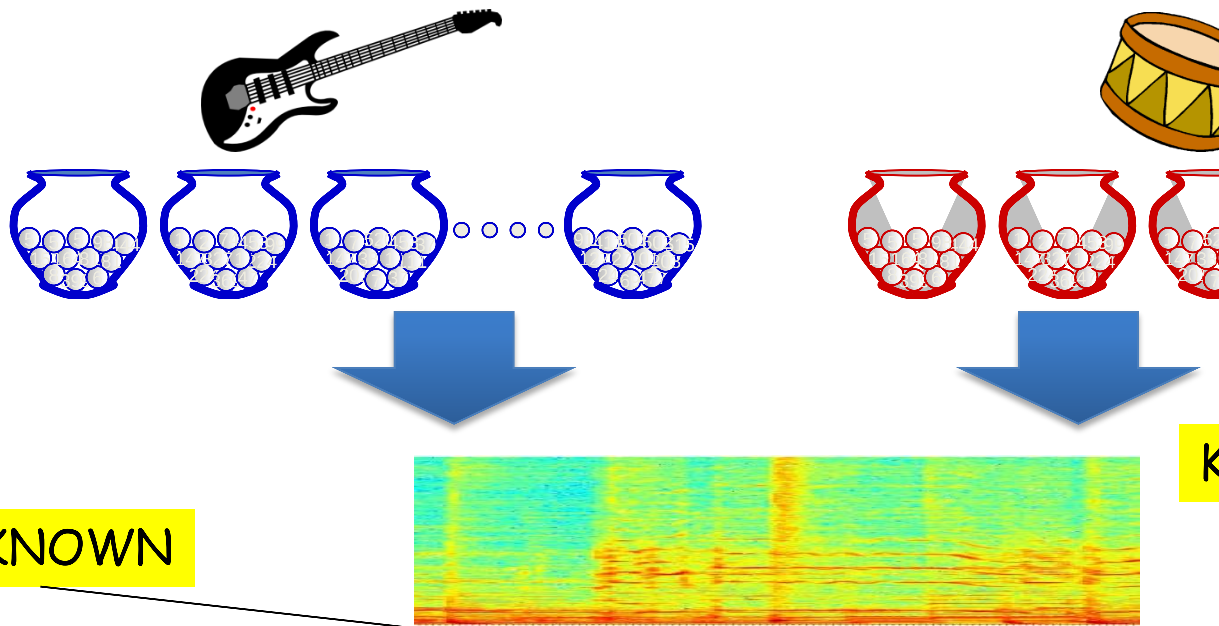
$$\boxed{P_t^{source2}(f) = \sum_{z\ for\ source2} P_t(z)P(f\,|\,z)}$$

Separating the Sources: Cleaner Solution

- 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-the source at frequency f
- A simple maximum a posteriori estimator

$$\hat{S}_{t,i}(f) = S_t(f)\frac{\sum_{z\ for\ source\ i} P_t(z)P(f\mid z)}{\sum_{all\ z} P_t(z)P(f\mid z)}$$

Semi-supervised separation: Example with two sources



UNKNOWN

KNOWN A PRIORI

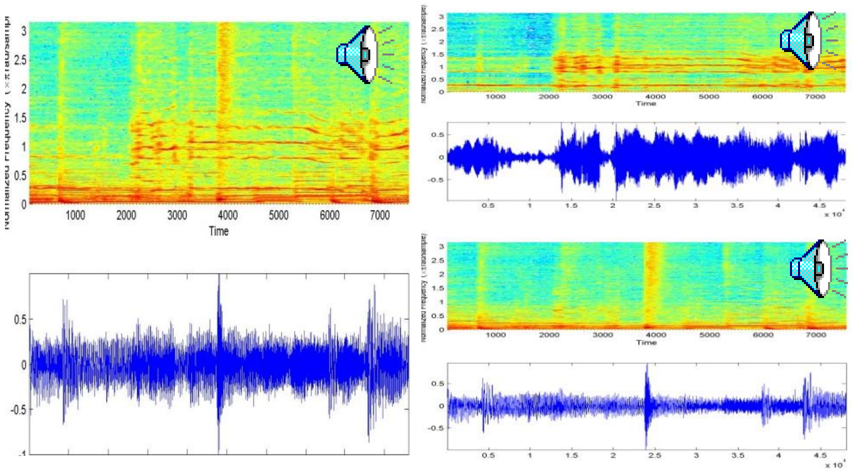
$$P_t(f) = \sum_{all\, z} P_t(z)P(f\,|\,z) = \sum_{z\ for\ source1} P_t(z)P(f\,|\,z) + \sum_{z\ for\ source2} P_t(z)P(f\,|\,z)$$

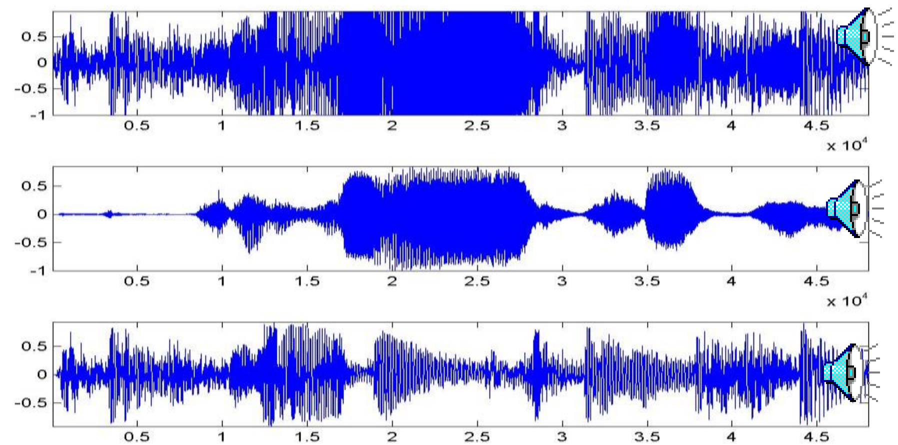
- Estimate from mixed signal (in addition to all $P_t(z)$)

$$P_t^{source1}(f) = \sum_{z\ for\ source1} P_t(z)P(f\,|\,z)$$

$$P_t^{source2}(f) = \sum_{z\ for\ source2} P_t(z)P(f\,|\,z)$$

Separating Mixed Signals: Examples





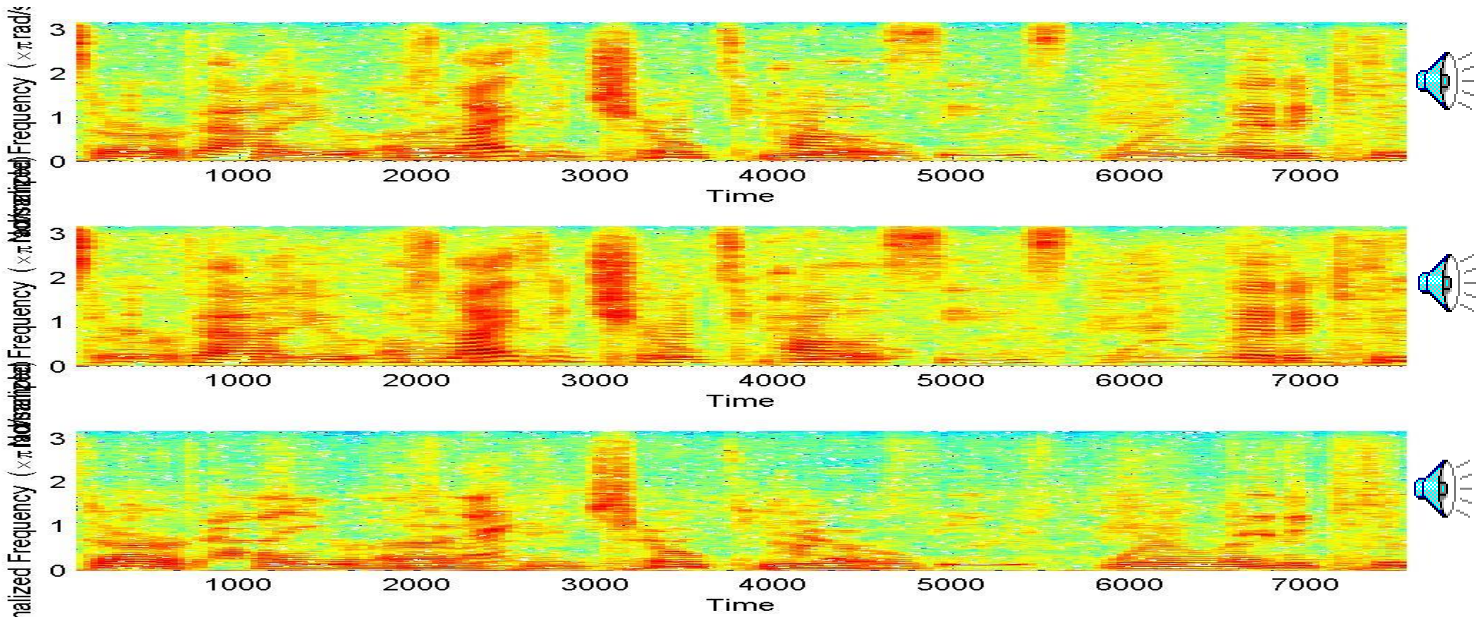
- "Raise my rent" by David Gilmour
- Background music "bases" learnt from 5-seconds of music-only segments within the song
- Lead guitar "bases" bases learnt from the rest of the song

- 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

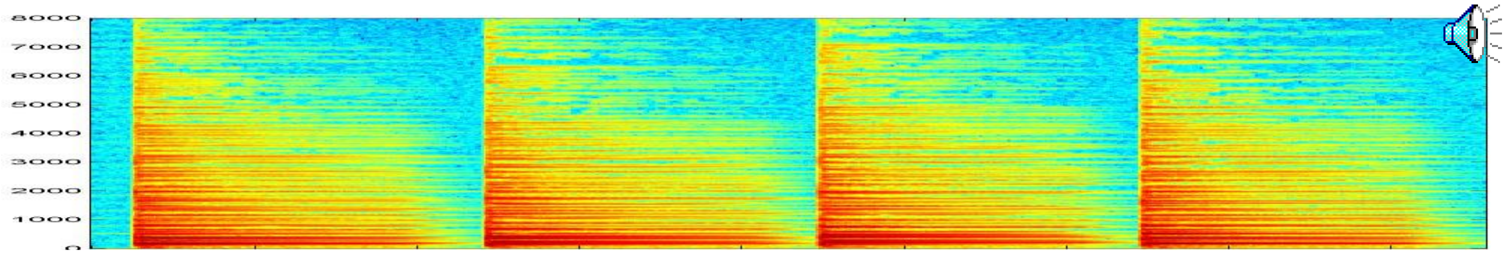
Can it be improved?

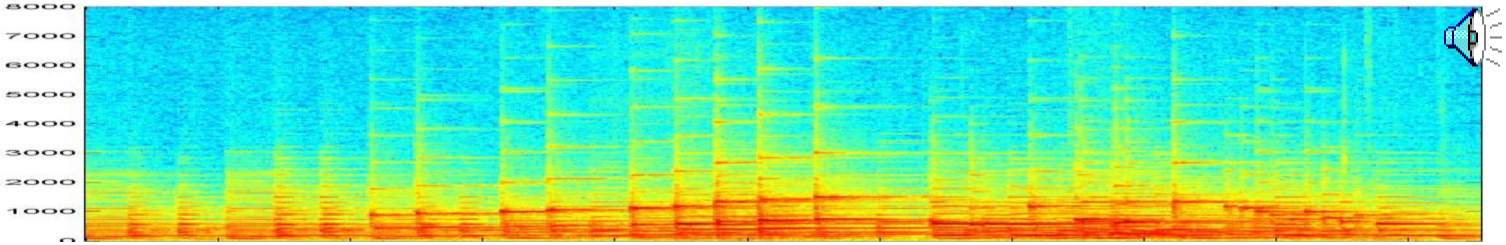
- Yes
- Tweaking
  - More training data per source
  - More bases per source
    - Typically about 40, but going up helps.
  - Adjusting FFT sizes and windows in the signal processing
- And / Or algorithmic improvements
  - Sparse overcomplete representations
  - Nearest-neighbor representations
  - Etc..

More on the topic

- Shift-invariant representations

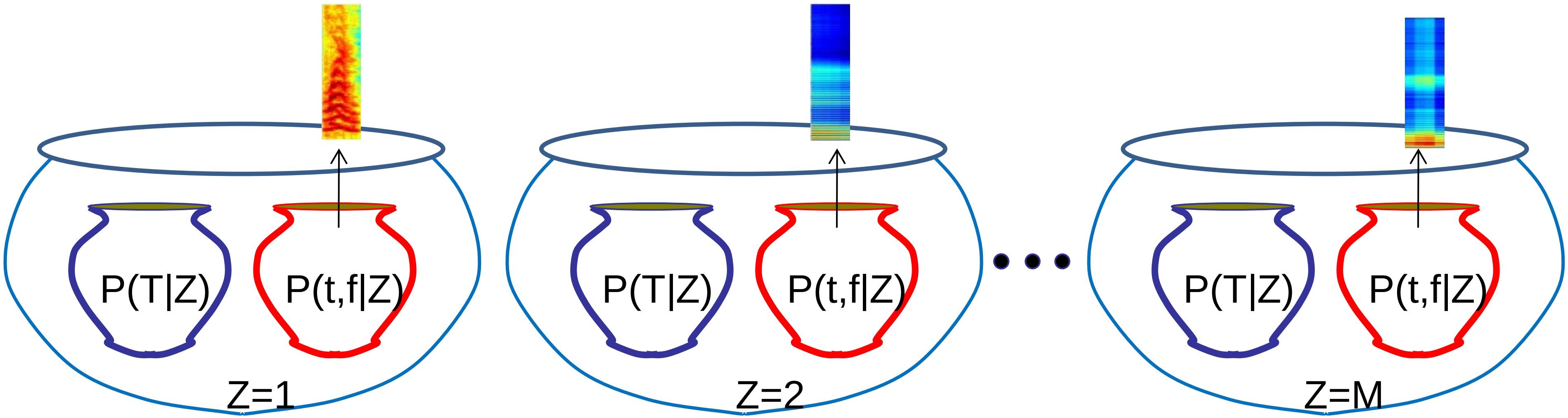
Patterns extend beyond a single frame





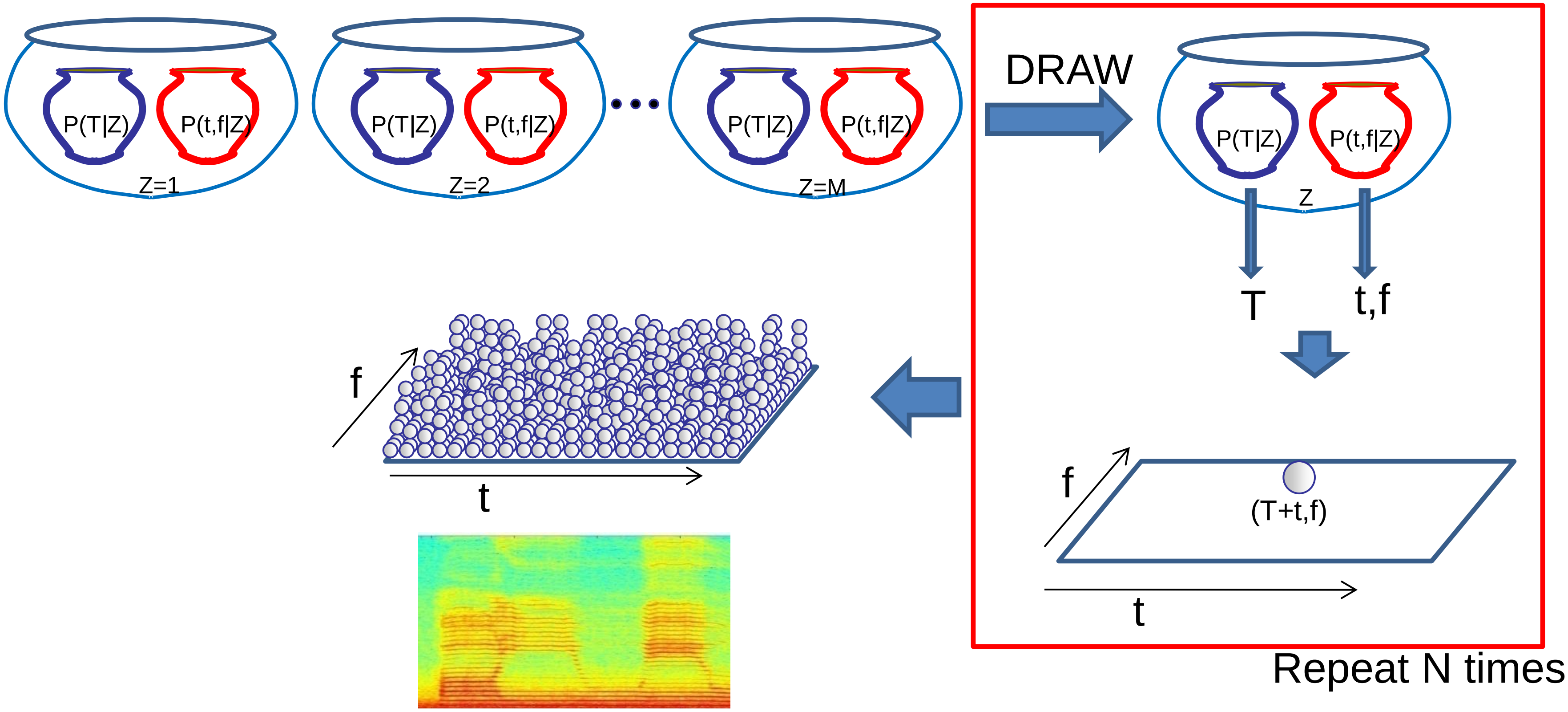
- Four bars from a music example
- The spectral patterns are actually patches
  - Not all frequencies fall off in time at the same rate
- The basic unit is a spectral patch, not a spectrum
- Extend model to consider this phenomenon

Shift-Invariant Model



- Employs bag of spectrograms model
- Each "super-urn" (*z*) has two sub urns
  - **One** suburn now stores a bi-variate distribution
    - Each ball has a (t,f) pair marked on it – the bases
  - Balls in the **other** suburn merely have a time "T" marked on them – the "location"

The shift-invariant model



$$P(t, f) = \sum_{Z} P(z) \sum_{T} P(T \mid z) P(T - t, f \mid z)$$

Estimating Parameters

- Maximum likelihood estimate follows fragmentation and counting strategy
- Two-step fragmentation
  - Each instance is fragmented into the super urns
  - The fragment in each super-urn is further fragmented into each time-shift
    - Since one can arrive at a given (t,f) by selecting any T from P(T|Z) and the appropriate shift t-T from P(t,f|Z)

Shift invariant model: Update Rules

- Given data (spectrogram) S(t,f)
- Initialize P(Z), P(T|Z), P(t,f | Z)
- Iterate

$$P(t,f,Z) = P(Z)\sum_{T} P(T|Z)P(t-T,f|Z) \qquad P(T,t,f|Z) = P(T|Z)P(t-T,f|Z)$$

$$P(Z|t,f) = \frac{P(t,f,Z)}{\sum_{Z'} P(t,f,Z')} \qquad P(T|Z,t,f) = \frac{P(T,t-T,f|Z)}{\sum_{T'} P(T',t-T',f|Z)}$$

Fragment

$$P(Z) = \frac{\sum_{t}\sum_{f} P(Z|t,f)S(t,f)}{\sum_{Z'}\sum_{t}\sum_{f} P(Z'|t,f)S(t,f)} \qquad P(T|Z) = \frac{\sum_{t}\sum_{f} P(Z|t,f)P(T|Z,t,f)S(t,f)}{\sum_{T'}\sum_{t}\sum_{f} P(Z|t,f)P(T'|Z,t,f)S(t,f)}$$

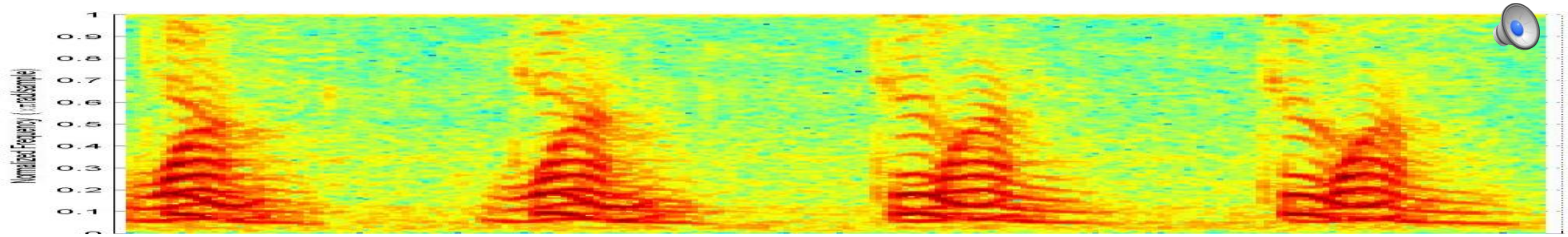
$$P(t,f|Z) = \frac{\sum_{T} P(Z|T,f)P(T-t|Z,T,f)S(T,f)}{\sum_{t'}\sum_{T} P(Z|T,f)P(T-t'|Z,T,f)S(T,f)}$$

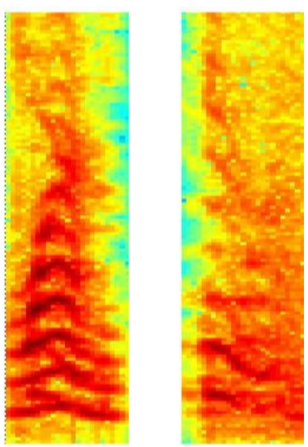
Count

An Example

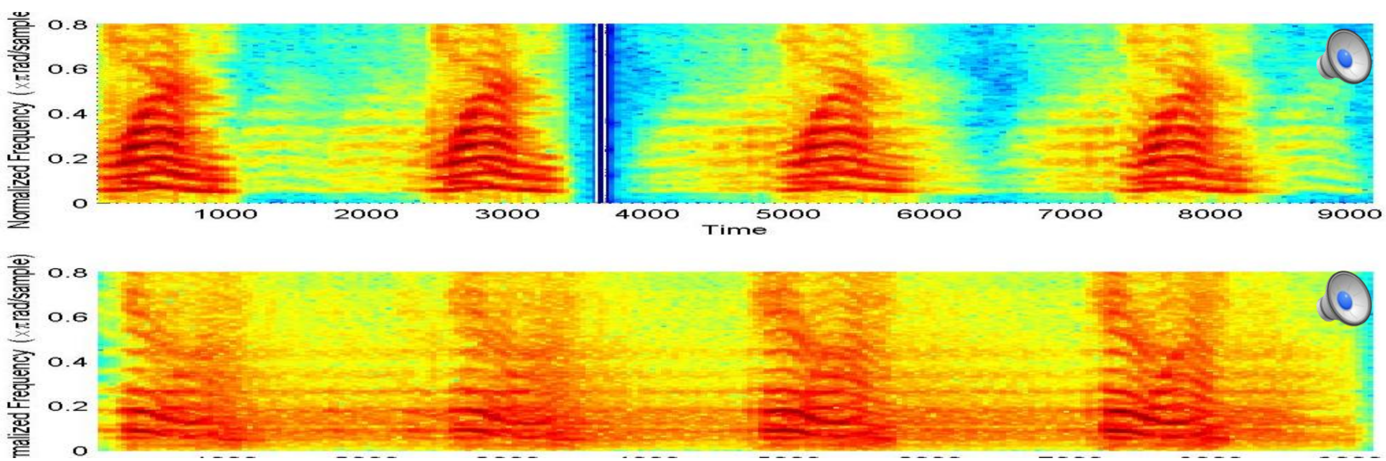
- Two distinct sounds occuring with different repetition rates within a signal

INPUT SPECTROGRAM



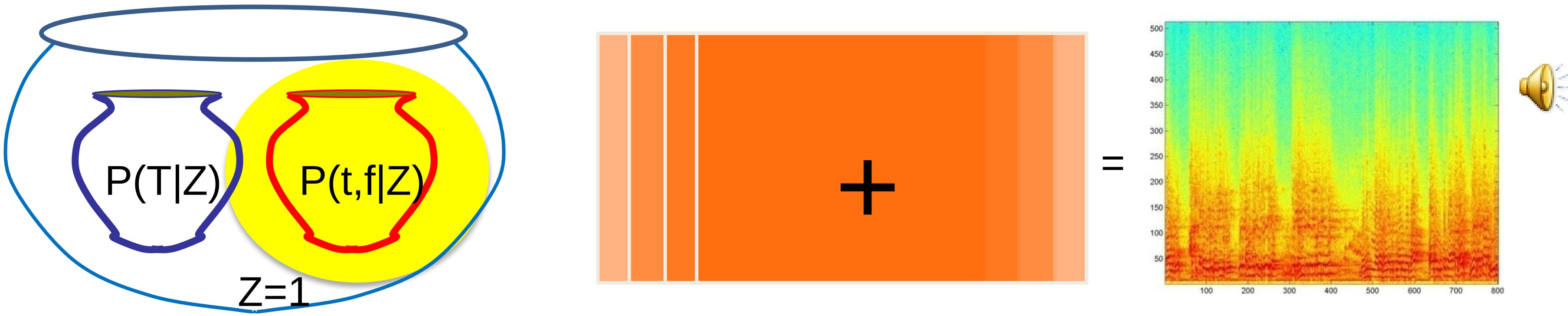


Discovered "patch" bases



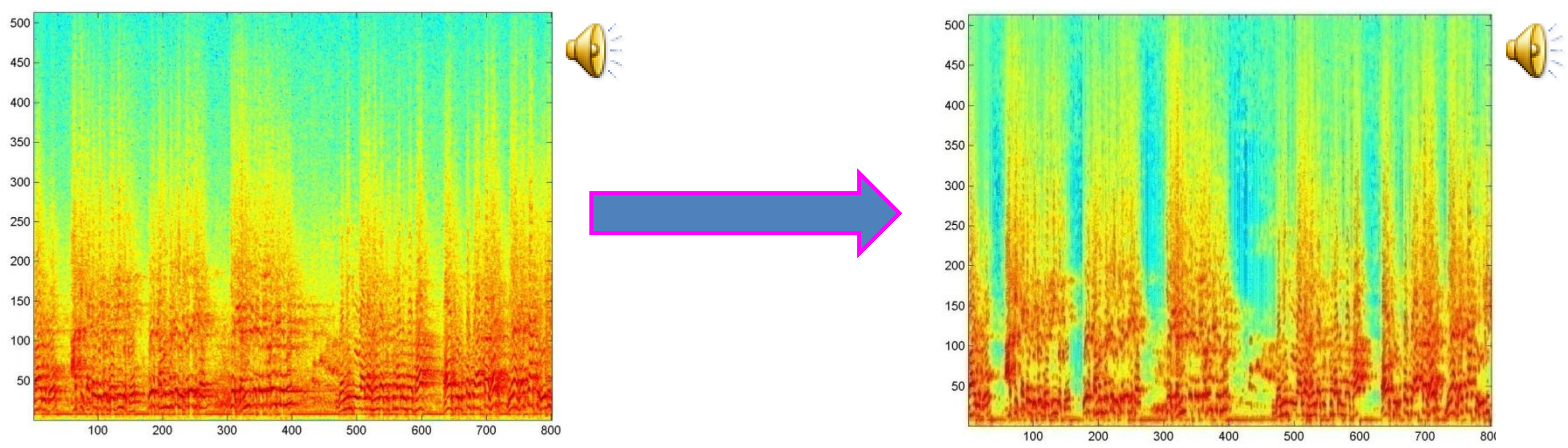
Contribution of individual bases to the recording

Another example: Dereverberation



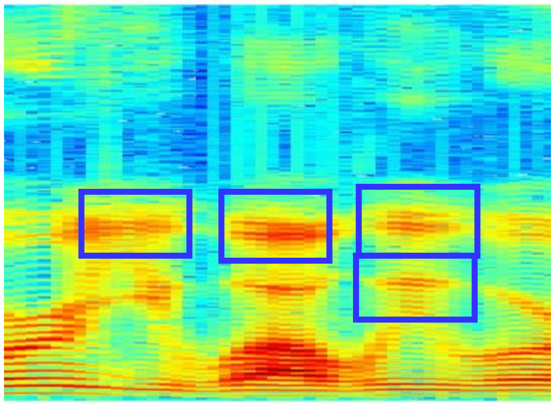
- Assume generation by a single latent variable
  - Super urn
- The t-f basis is the "clean" spectrogram

Dereverberation: an example



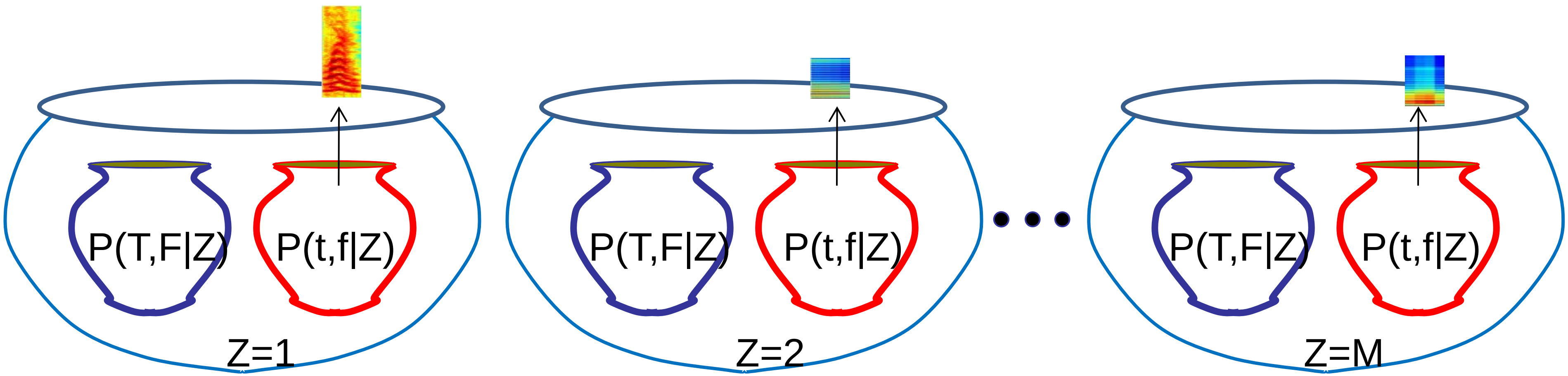
- "Basis" spectrum must be made sparse for effectiveness
- Dereverberation of gamma-tone spectrograms is also particularly effective for speech recognition

Shift-Invariance in Two dimensions



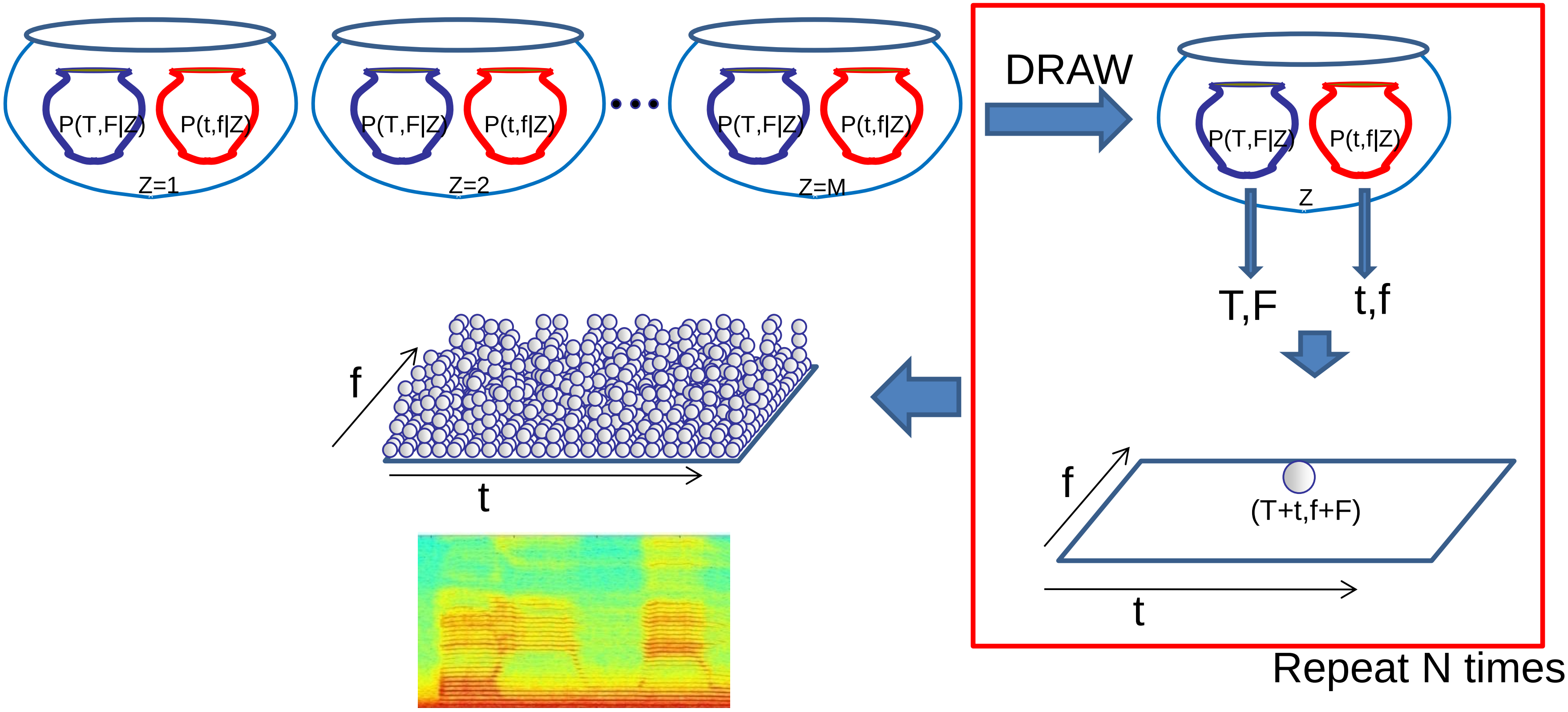
- Patterns may be substructures
  - Repeating patterns that may occur anywhere
    - Not just in the same frequency or time location
    - More apparent in image data

The two-D Shift-Invariant Model



- Both sub-pots are distributions over (T,F) pairs
  - One subpot represents the basic pattern
    - Basis
  - The other subpot represents the *location*

The shift-invariant model



$$P(t,f) = \sum_{Z} P(z) \sum_{T} \sum_{F} P(T,F \mid z) P(T-t, f-F \mid z)$$

Two-D Shift Invariance: Estimation

- Fragment and count strategy
- Fragment into superpots, but also into each *T and F*
  - Since a given (t,f) can be obtained from any (T,F)

$$P(t,f,Z) = P(Z)\sum_{T,F} P(T,F|Z)P(t-T,f-F|Z) \qquad P(T,F,t,f|Z) = P(T,F|Z)P(t-T,f-F|Z)$$

$$P(Z|t,f) = \frac{P(t,f,Z)}{\sum_{Z'} P(t,f,Z')} \qquad P(T,F|Z,t,f) = \frac{P(T,F,t-T,f-F|Z)}{\sum_{T',F'} P(T',F',t-T',f-F'|Z)}$$

Fragment

$$P(Z) = \frac{\sum_t \sum_f P(Z|t,f)S(t,f)}{\sum_{Z'}\sum_t \sum_f P(Z'|t,f)S(t,f)} \qquad P(T,F|Z) = \frac{\sum_t \sum_f P(Z|t,f)P(T,F|Z,t,f)S(t,f)}{\sum_{T'}\sum_{F'}\sum_t \sum_f P(Z|t,f)P(T',F'|Z,t,f)S(t,f)}$$

$$P(t,f|Z) = \frac{\sum_{T,F} P(Z|T,F)P(T-t,F-f|Z,T,F)S(T,F)}{\sum_{t',f'}\sum_{T,F} P(Z|T,F)P(T-t',F-f'|Z,T,F)S(T,F)}$$

Count

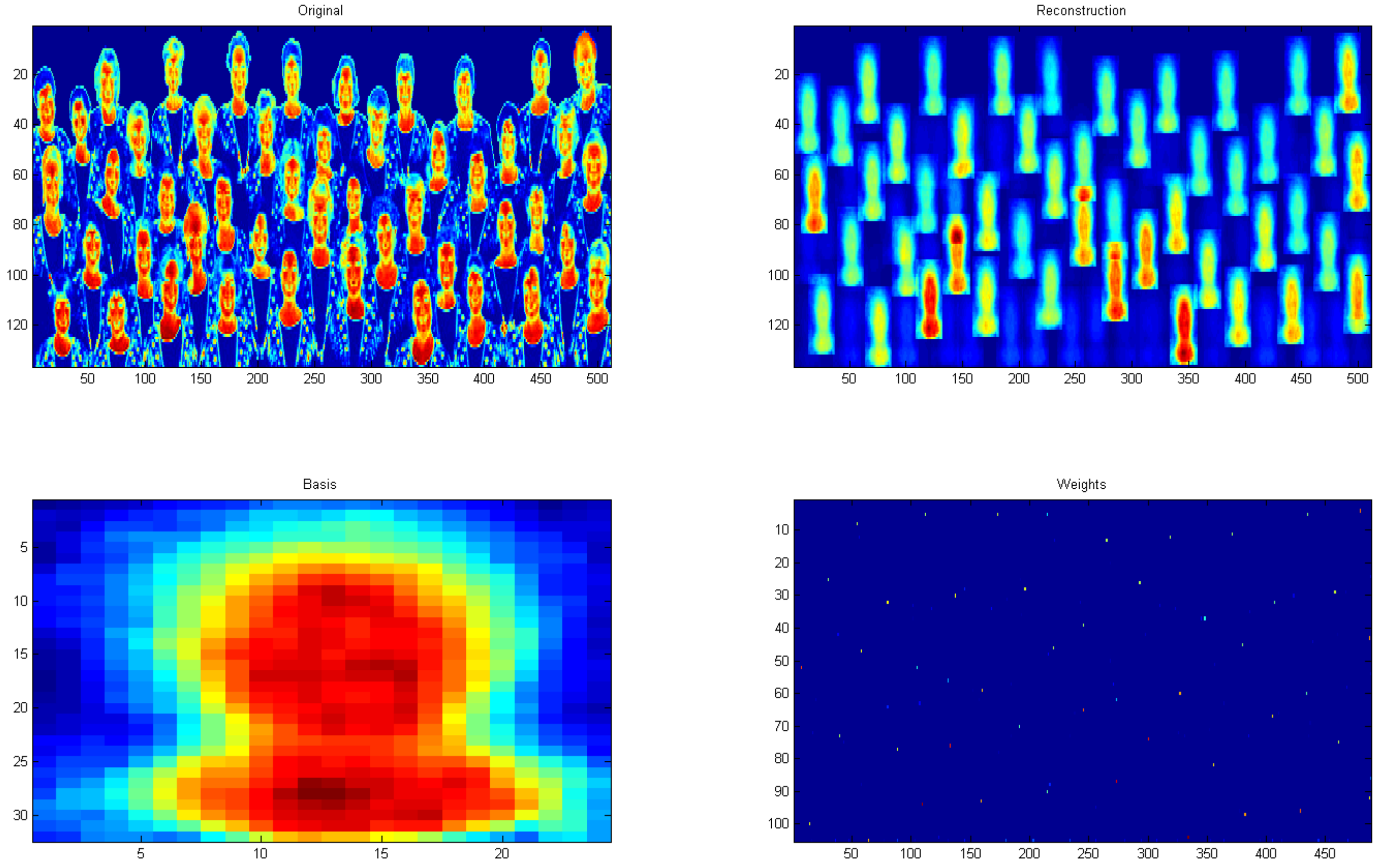
Shift-Invariance: Comments

- P(T,F|Z) and P(t,f|Z) are symmetric
  - Cannot control which of them learns patterns and which the locations
- Answer: Constraints
  - Constrain the size of P(t,f|Z)
    - I.e. the size of the basic patch
  - Other tricks – e.g. sparsity

Shift-Invariance in Many Dimensions

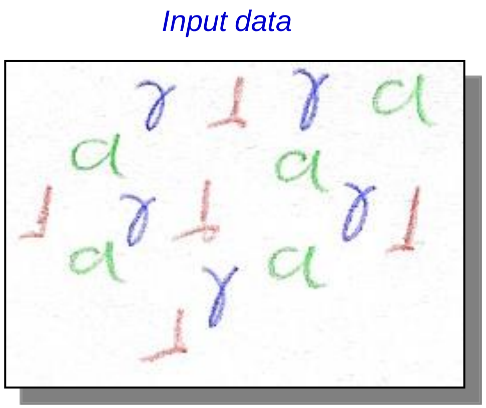
- The generic notion of "shift-invariance" can be extended to multivariate data
  - Not just two-D data like images and spectrograms
- Shift invariance can be applied to any subset of variables

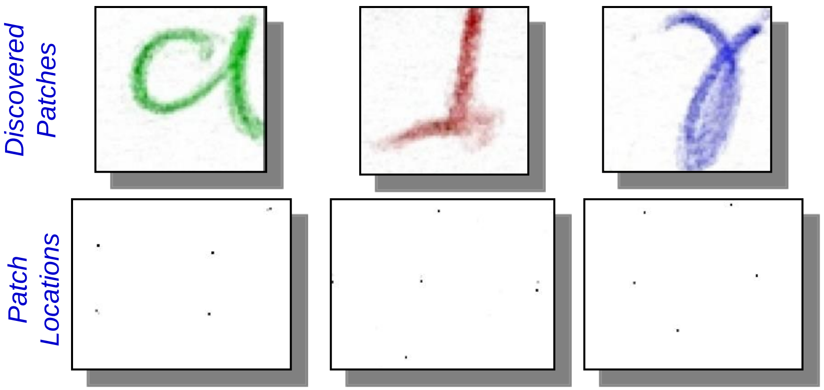
Example: 2-D shift invariance



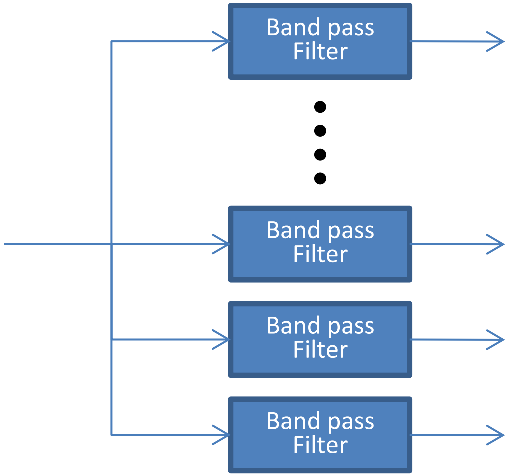
Example: 3-D shift invariance

- The original figure has multiple handwritten renderings of three characters
  – In different colours
- The algorithm learns the three characters and identifies their locations in the figure



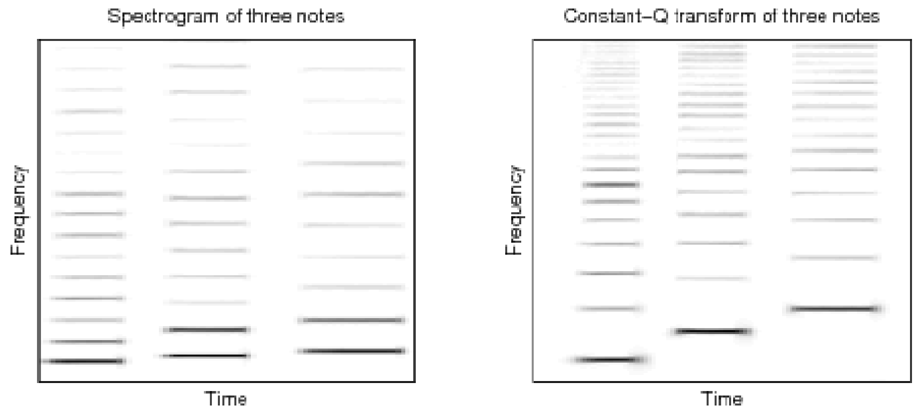


The constant Q transform



- Spectrographic analysis with a bank of constant Q filters
  - The bandwidth of filters increases with center frequency.
  - The spacing between filter center frequencies increases with frequency
    - Logarithmic spacing

Constant Q representation of Speech

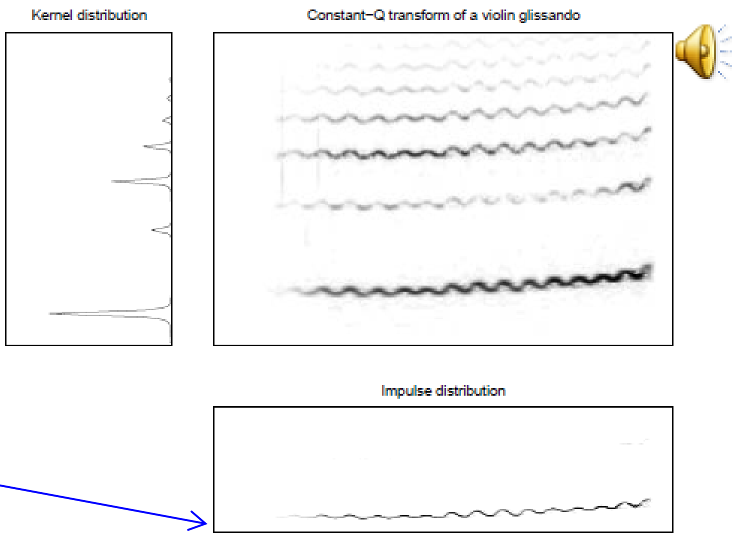


- Energy at the output of a bank of filters with logarithmically spaced center frequencies
  - Like a spectrogram with non-linear frequency axis
- Changes in pitch become vertical translations of spectrogram
  - Different notes of an instrument will have the same patterns at different vertical locations

Pitch Tracking

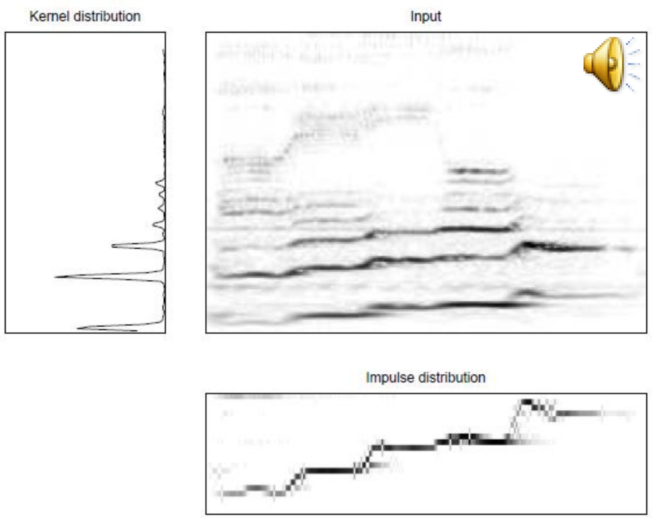
$$P(t,f)=\sum_{z}P(z)\sum_{T,F}P_s(T,F\,|\,z)P(t-T,f-F\,|\,z)$$

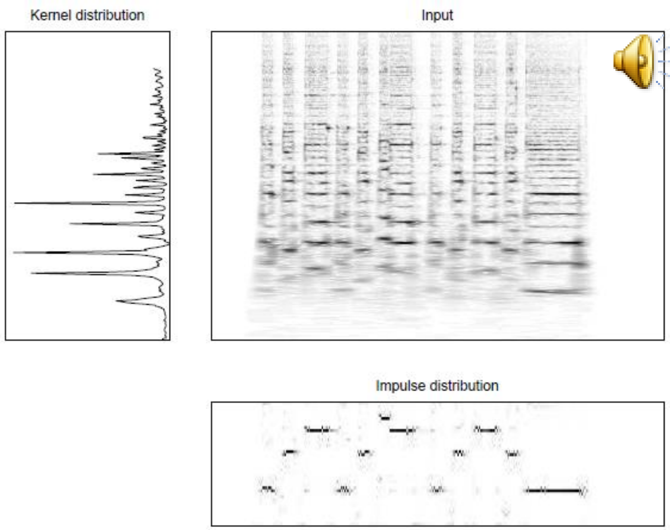
$$P(t,f)=\sum_{F}P(t,F)P(f-F)$$



- Changing pitch becomes a vertical shift in the location of a basis
- The constant-Q spectrogram is modeled as a single pattern modulated by a vertical shift
  - P(f) is the "Kernel" shown to the left

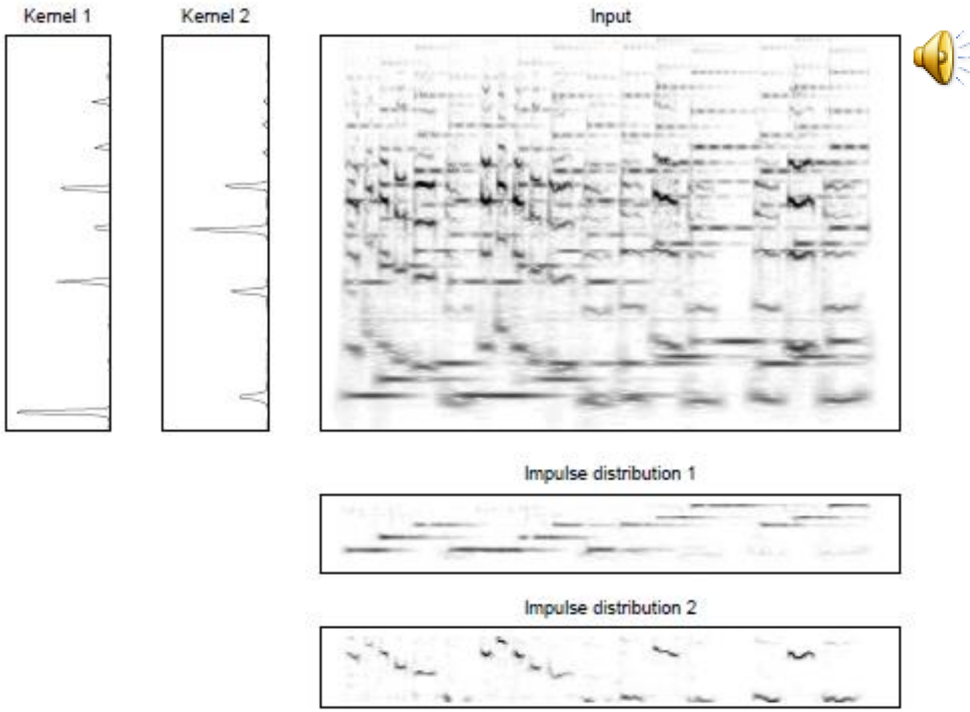
Pitch Tracking





- Left: A vocalized "song"
- Right: Chord sequence
- "Impulse" distribution captures the "melody"!

Pitch Tracking



- Having more than one basis (z) allows simultaneous pitch tracking of multiple sources
- Example: A voice and an instrument overlaid
  - The “impulse” distribution shows pitch of both separately

In Conclusion

- Surprising use of EM for estimation of latent structure for audio analysis
- Various extensions
  - Sparse estimation
  - Exemplar based methods..
- Related deeply to non-negative matrix factorization
  - TBD..