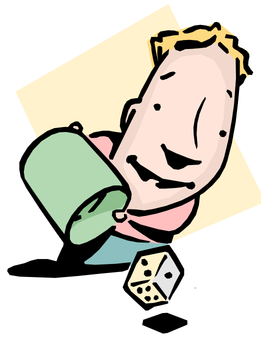
Latent Variable Models and Signal Separation

Class 13. 07 Oct 2009

Separating Mixed Signals an example

- "Raise my rent" by David Gilmour

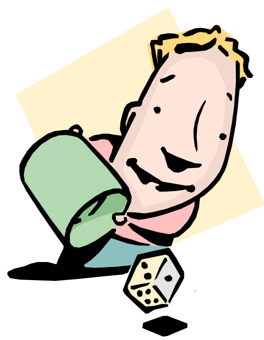
A Thought Experiment



6 3 1 5 4 1 2 4 …

- A person shoots dice repeatedly
- The dice are loaded
- You may observe the series of outcomes
- **After observing the outcomes for some time, you can form a good idea of how the dice is loaded**
  - Figure out what the probabilities of the various numbers are for dice
- P(number) = count(number)/sum(rolls)
- This is a *maximum likelihood* estimate
  - Estimate that makes the observed sequence of numbers most probable

A Thought Experiment



6 3 1 5 4 1 2 4 …

4 4 1 6 3 2 1 2 …

- Two persons shoot dice repeatedly
- The dice are loaded
  - The dice are differently loaded for the two of them
- You may observe the series of outcomes for both persons
- **After observing the outcomes for some time, you can form a good idea of how each of the two dice is loaded**
  - Figure out what the probabilities of the various numbers are on each set dice

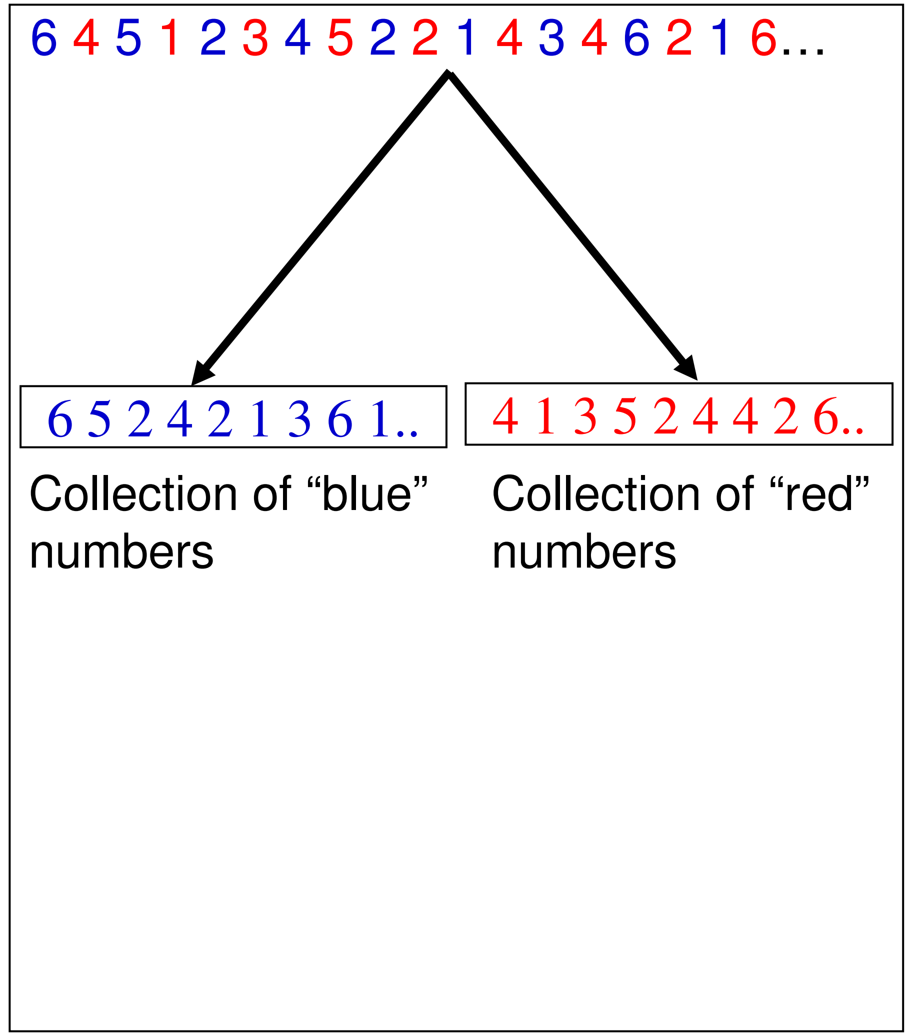
Estimating Probabilities

- Observation: Observe the sequence of numbers from the two shooters
  - As indicated by the colors, we know who rolled what number

6 4 5 1 2 3 4 5 2 2 1 4 3 4 6 2 1 6…

Estimating Probabilities

- Observation: Observe the sequence of numbers from the two shooters
  - As indicated by the colors, we know who rolled what number
- Segregation: Separate the blue observations from the red



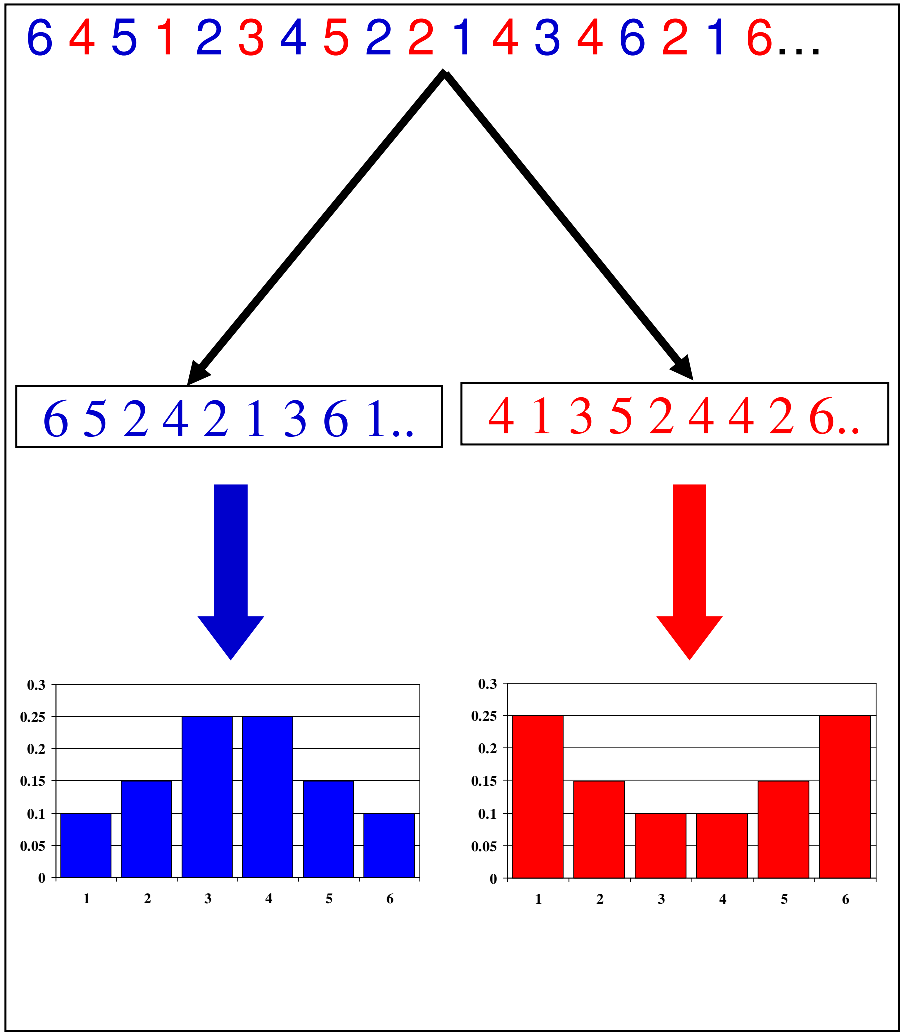
Estimating Probabilities

- Observation: Observe the sequence of numbers from the two shooters
  - As indicated by the colors, we know who rolled what number

- Segregation: Separate the blue observations from the red

- From each set compute probabilities for each of the 6 possible outcomes

$$P(number) = \frac{\text{no. of times number was rolled}}{\text{total number of observed rolls}}$$



A Thought Experiment

6 3 1 5 4 1 2 4 …

4 4 1 6 3 2 1 2 …

- Now imagine that you cannot observe the dice yourself
- Instead there is a "caller" who randomly calls out the outcomes of the rolls
  - 40% of the time he calls out the number from the left shooter, and 60% of the time, the one from the right (and you know this)
- At any time, you do not know which of the two he is calling out
- How do you now determine the probability distributions for the two sets of dice?

A Thought Experiment

6 3 1 5 4 1 2 4 …

4 4 1 6 3 2 1 2 …

- Now imagine that you cannot observe the dice yourself
- Instead there is a "caller" who randomly calls out the outcomes of the rolls
  - 40% of the time he calls out the number from the left shooter, and 60% of the time, the one from the right (and you know this)
- At any time, you do not know which of the two he is calling out
- How do you now determine the probability distributions for the two sets of dice?
- If you do not even know what fraction of time the blue numbers are called, and what fraction are red?

Probabilities to Estimate

- The caller will call out a number 6 in any given callout IF
  - He selects “RED”, and the Red die rolls the number 6
  - OR
  - He selects “BLUE” and the Blue die rolls the number 6

- So the probability that he will call out 6 is:
  - Prob(RED)*P(6 | RED) + Prob(BLUE)*P(6|BLUE)

- More generically
  - P(X) = P(Red)P(X|Red) + P(Blue)P(X|Blue)

- What we must estimate from the sequence of numbers called out
  - P(RED) and P(BLUE) – the probabilities that he will select either die
  - P(X|RED) and P(X|BLUE) – the probability distribution of the numbers 1-6 for both dice!

Multinomials and Mixture Multinomials

- A probability distribution over a collection of items, each of which may be drawn in any draw is a *Multinomial*

$$P(X : X \text{ belongs to a discrete set}) = P(X)$$

- A probability distribution that *combines* (or *mixes*) draws from multiple multinomials is a *mixture multinomial*

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

Mixture weights — Component multinomials

Expectation Maximization

- It is possible to estimate all parameters in this setup using the Expectation Maximization (or EM) algorithm

- First described in a landmark paper by Dempster, Laird and Rubin
  - Maximum Likelihood Estimation from incomplete data, via the EM Algorithm, Journal of the Royal Statistical Society, Series B, 1977

- Much work on the algorithm since then
  - McLachlan, Bashford, …….

- The principles behind the algorithm existed for several years prior to the landmark paper, however.

EM results in maximum likelihood estimates

$$P(X) = \sum_Z P(Z)P(X \mid Z)$$

- P(X) = P(O==X) is the probability that any observation *O* will take value *X*
  - *i.e.* That the probability that number rolled is X
- EM estimates of P(Z) and P(X|Z) are such that:

  $P(O_1, O_2, ..) = P(O_1)P(O_2)P(O_3)..$

  is maximized
- This too is a maximum-likelihood solution

Expectation Maximization

- Iterative solution
- Get some initial estimates for all parameters
  - Dice shooter example: This includes probability distributions for dice AND the probability with which the caller selects the dice
- Two steps that are iterated:
  - ***Expectation Step:*** Estimate statistically, the values of ***unseen*** variables
  - ***Maximization Step:*** Using the estimated values of the unseen variables as truth, estimates of the model parameters

Expectation Maximization: Terminology

- Hidden variable: *Z*
  - Dice: The identity of the shooter whose dice roll has been called out
- *A priori* probability distribution of hidden variable *P*(*Z*)
  - Dice: Probability that the caller will call the red shooter; probability that he will call the blue shooter
    - For what fraction of a very large number of calls he calls the red shooter
- Observed data: *X*
  - The numbers called out
- Parameters that could be estimated, if the hidden variable was known: P(*X* | *Z*) and P(Z)
  - Dice: For the dice example, these would be the probabilities of the numbers 1 – 6 for each shooter (6 values for each shooter, 12 in all)
  - And, the probability that the caller selects either die

Expectation Maximization

- If we knew the value of *Z* for every observation, we could estimate P(*X*|*Z*)
  - If we knew which shooter rolled each number, we could estimate the probability of the dice for both shooters
- Unfortunately, we do not know *Z* – it is hidden from us!
- Reverse the problem: try to estimate *Z* after having seen *X*
  - Guess who rolled the dice ***from the number***
  - If the blue shooter shoots “4” much more often than the red shooter, and if the caller calls out “4”, then the caller has probably called out the blue shooter
  - This is an *a posteriori* estimate: estimation posterior to the observation

Expectation Maximization

- The Expectation step of EM attempts to estimate the hidden variable *Z* from the observed data *X*

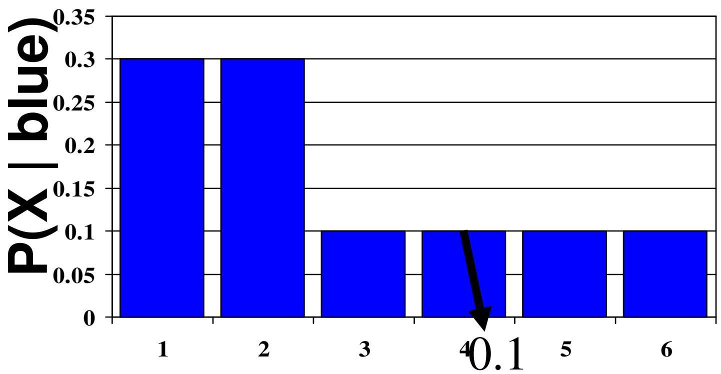
- Since we can usually not be certain of the estimate for *Z*, *Z* is probabilistically estimated:
  - Instead of saying "The caller called the Blue shooter", we say "After observing that the caller called a 4, we estimate that he may have called the blue shooter with probability 0.667, and the red shooter with probability 0.333
  - The post observation estimates of the probabilities of the various values of *Z* are called *a posteriori* probabilities

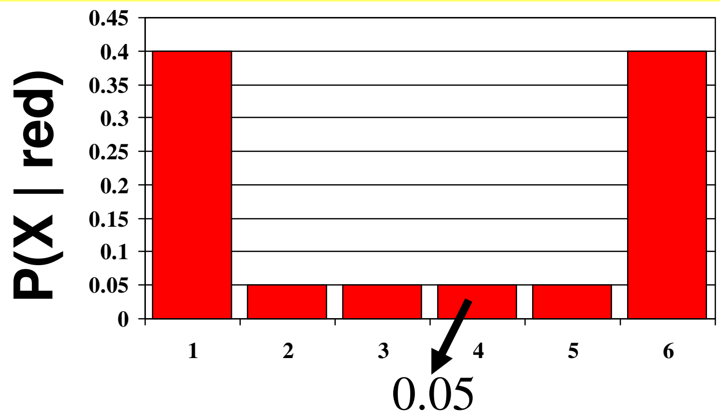
- The *a posteriori* probabilities of the various values of *Z* are computed using Bayes' rule:

$$P(Z \mid X) = \frac{P(X \mid Z)P(Z)}{P(X)} = CP(X \mid Z)P(Z)$$

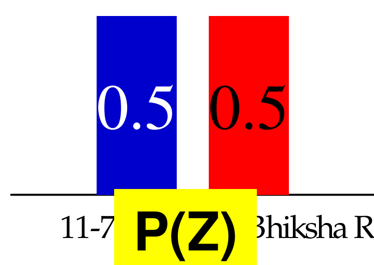
Expectation Maximization

- Hypothetical Dice Shooter Example:
- We obtain an initial estimate for the probability distribution of the two sets of dice (somehow):





- We obtain an initial estimate for the probability with which the caller calls out the two shooters (somehow)



Expectation Maximization

- Hypothetical Dice Shooter Example:
- We have an initial estimate:
  - caller calls blue 0.5 of the time, and red 0.5 of the time
  - Probability of "4" for blue die is 0.1, for red die is 0.05"
  - Caller has just called out 4
- Observation $X = 4$. From initial estimates:
  - $P(X \mid Z=\text{red}) = 0.1$; $P(X \mid Z=\text{blue}) = 0.05$
  - $P(Z=\text{red}) = 0.5$; $P(Z=\text{blue}) = 0.5$

$$P(red \mid X = 4) = CP(X = 4 \mid Z = red)P(Z = red) = C \cdot 0.05 \cdot 0.5 = C0.025$$

$$P(blue \mid X = 4) = CP(X = 4 \mid Z = blue)P(Z = blue) = C \cdot 0.1 \cdot 0.5 = C0.05$$

$$\text{Normalizing}: \; P(red \mid X = 4) = 0.33; \quad P(blue \mid X = 4) = 0.67$$

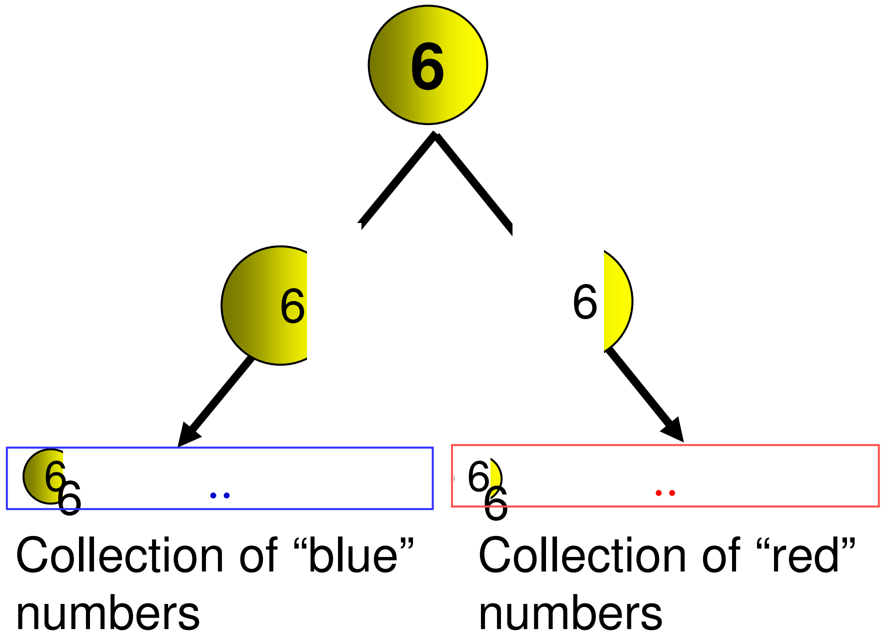
Expectation Maximization

6 4 5 1 2 3 4 5 2 2 1 4 3 4 6 2 1 6

- For each observation O==X, $P(Z \mid X)$ must be computed for every value of Z and for every observation O
- In the dice example, we must compute both $P(\text{red} \mid X)$ and $P(\text{blue} \mid X)$ for every observation O==X
  - An observation here is a called out roll of the dice

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

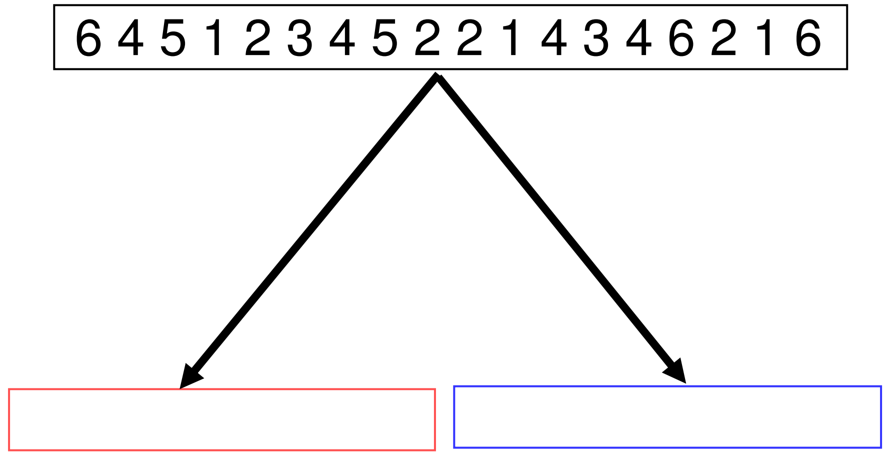
Expectation Maximization



- Each call is “fragmented”
  - Fragment sizes are proportional to the a posteriori probabilities of the colors
    - *P(Z|X)*
- The fragments are added to the collections associated with the different dice
  - So a fragment of *every* observation ends up in the collection for any dice

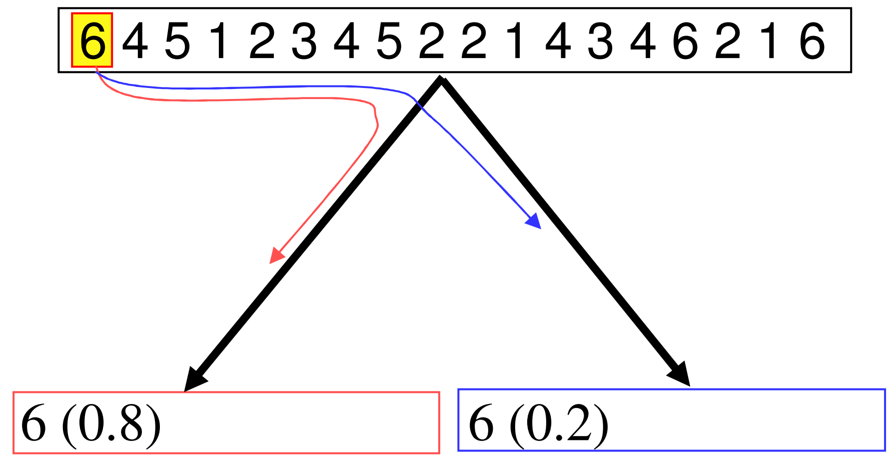
Expectation Maximization

- Every observed roll of the dice contributes to both "Red" and "Blue"



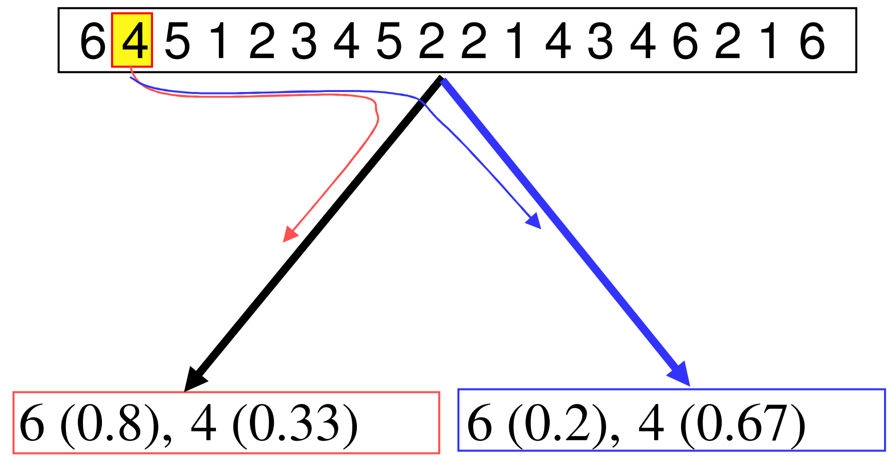
Expectation Maximization

- Every observed roll of the dice contributes to both "Red" and "Blue"



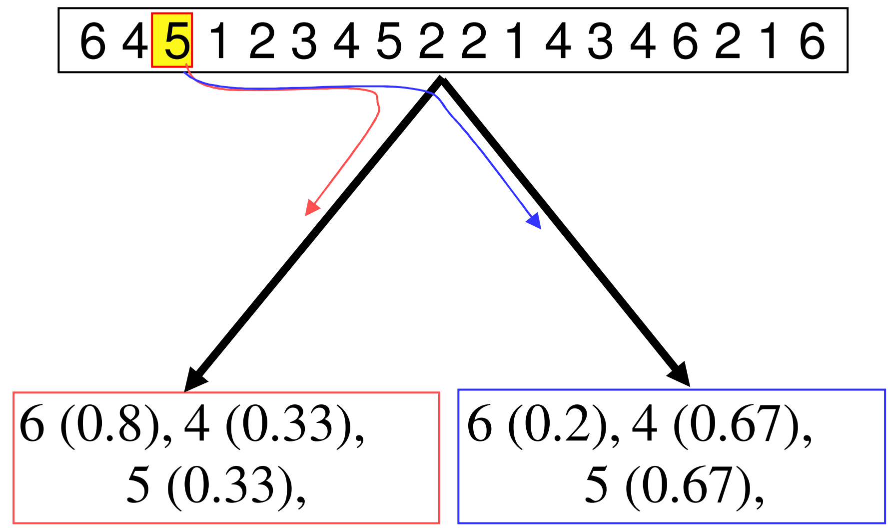
Expectation Maximization

- Every observed roll of the dice contributes to both "Red" and "Blue"



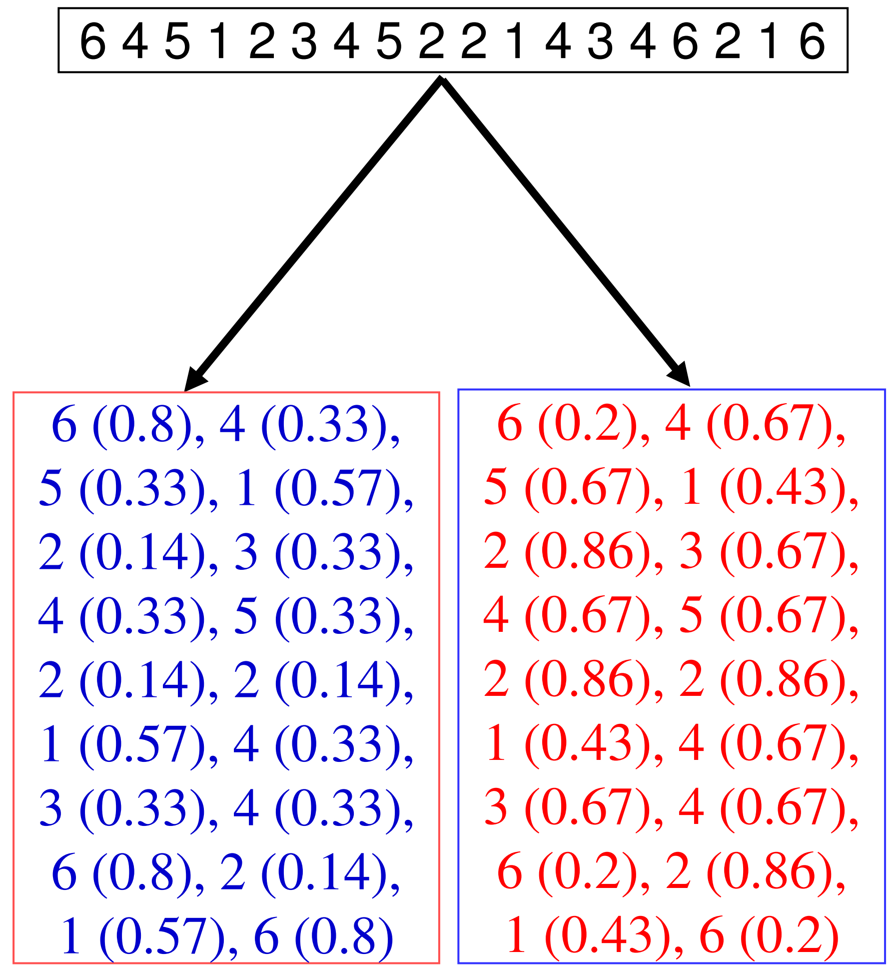
Expectation Maximization

- Every observed roll of the dice contributes to both “Red” and “Blue”



Expectation Maximization

- Every observed roll of the dice contributes to both “Red” and “Blue”



Expectation Maximization

- Every observed roll of the dice contributes to both "Red" and "Blue"
- Total count for "Red" is the sum of all the posterior probabilities in the red column
  - 7.31
- Total count for "Blue" is the sum of all the posterior probabilities in the blue column
  - 10.69
  - Note: 10.69 + 7.31 = 18 = the total number of instances

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

Expectation Maximization

- Total count for “Red” : 7.31
- Red:
  - Total count for 1: 1.71

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

Expectation Maximization

- Total count for "Red" : 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56

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

Expectation Maximization

- Total count for “Red” : 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66

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

Expectation Maximization

- Total count for "Red" : 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66
  - Total count for 4: 1.32

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

Expectation Maximization

- Total count for “Red” : 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66
  - Total count for 4: 1.32
  - Total count for 5: 0.66

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

Expectation Maximization

- Total count for "Red" : 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66
  - Total count for 4: 1.32
  - Total count for 5: 0.66
  - Total count for 6: 2.4

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

Expectation Maximization

- Total count for "Red" : 7.31
- Red:
  - Total count for 1: 1.71
  - Total count for 2: 0.56
  - Total count for 3: 0.66
  - Total count for 4: 1.32
  - Total count for 5: 0.66
  - Total count for 6: 2.4

- **Updated probability of Red dice:**
  - 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

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

Expectation Maximization

- Total count for “Blue” : 10.69
- Blue:
  - Total count for 1: 1.29

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

Expectation Maximization

- Total count for “Blue” : 10.69
- Blue:
  - Total count for 1: 1.29
  - Total count for 2: 3.44

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

Expectation Maximization

- Total count for “Blue” : 10.69
- Blue:
  - Total count for 1: 1.29
  - Total count for 2: 3.44
  - Total count for 3: 1.34

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

Expectation Maximization

- Total count for "Blue" : 10.69
- Blue:
  - Total count for 1: 1.29
  - Total count for 2: 3.44
  - Total count for 3: 1.34
  - Total count for 4: 2.68

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

Expectation Maximization

- Total count for "Blue" : 10.69
- Blue:
  - Total count for 1: 1.29
  - Total count for 2: 3.44
  - Total count for 3: 1.34
  - Total count for 4: 2.68
  - Total count for 5: 1.34

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

Expectation Maximization

- Total count for “Blue” : 10.69
- Blue:
  - Total count for 1: 1.29
  - Total count for 2: 3.44
  - Total count for 3: 1.34
  - Total count for 4: 2.68
  - Total count for 5: 1.34
  - Total count for 6: 0.6

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

Expectation Maximization

- Total count for "Blue" : 10.69
- Blue:
  - Total count for 1: 1.29
  - Total count for 2: 3.44
  - Total count for 3: 1.34
  - Total count for 4: 2.68
  - Total count for 5: 1.34
  - Total count for 6: 0.6
- **Updated probability of Blue dice:**
  - 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

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

Expectation Maximization

- Total count for “Red” : 7.31
- Total count for “Blue” : 10.69
- Total instances = 18
  - Note 7.31+10.69 = 18
- We also revise our estimate for the probability that the caller calls out Red or Blue
  - i.e the fraction of times that he calls Red and the fraction of times he calls Blue

- 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

The updated values

- Probability of Red dice:
  - 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 dice:
  - 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
- **THE UPDATED VALUES CAN BE USED TO REPEAT THE PROCESS. ESTIMATION IS AN ITERATIVE PROCESS**

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

The Dice Shooter Example

6 3 1 5 4 1 2 4 …

4 4 1 6 3 2 1 2 …

1. Initialize $P(Z)$, $P(X \mid Z)$
2. Estimate $P(Z \mid X)$ for each Z, for each called out number
   - Associate X with each value of Z, with weight $P(Z \mid X)$
3. Re-estimate $P(X \mid Z)$ for every value of X and Z
4. Re-estimate $P(Z)$
5. If not converged, return to 2

In Squiggles

- Given a sequence of observations O_1, O_2, ..
  - N_X is the number of observations of color X
- Initialize P(Z), P(X|Z) for dice Z and numbers X
- Iterate:
  - For each number X:

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

  - Update:

$$P(X|Z) = \frac{\sum_{O \text{ such that } O==X} P(Z|X)}{\sum_{Z'} \sum_{O \text{ such that } O==X} P(Z'|X)} = \frac{N_X P(Z|X)}{\sum_{Z'} N_X P(Z'|X)}$$

$$P(Z) = \frac{\sum_X N_X P(Z|X)}{\sum_{Z'} \sum_X N_X P(Z|X)}$$

Expectation Maximization

- The EM algorithm is used whenever proper statistical analysis of a phenomenon requires the knowledge of a hidden or missing variable (or a set of hidden/missing variables)
  - The hidden variable is often called a “latent” variable

- Some examples:
  - Estimating mixtures of distributions
    - Only data are observed. The individual distributions and mixing proportions must both be learnt.
  - Estimating the distribution of data, when some attributes are missing
  - Estimating the dynamics of a system, based only on observations that may be a complex function of system state

The Mad Caller

- The EM algorithm will give us one of many solutions, all equally valid!
  - The probability of 6 being called out:

$$P(6) = aP(6 \mid red) + bP(6 \mid blue) = aP_r + bP_b$$

    - Assigns P_r as the probability of 6 for the red die
    - Assigns P_b as the probability of 6 for the blue die

  - The following too is a valid solution [FIX]

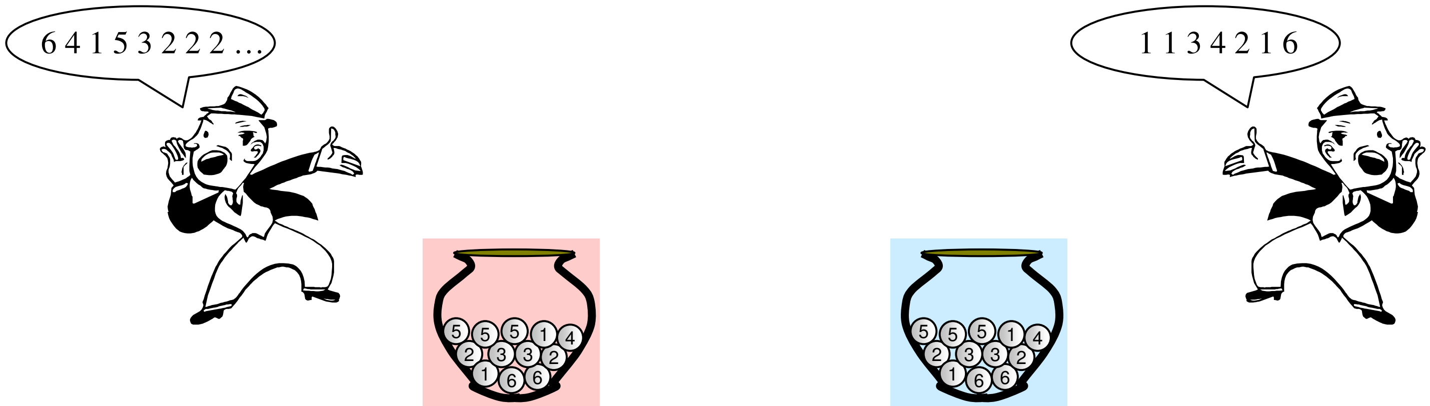
$$P(6) = 1.0\,(aP_r + bP_b) + 0.0 anything$$

    - Assigns 1.0 as the a priori probability of the red die
    - Assigns 0.0 as the probability of the blue die
- The solution is NOT unique

A mild shift of metaphor

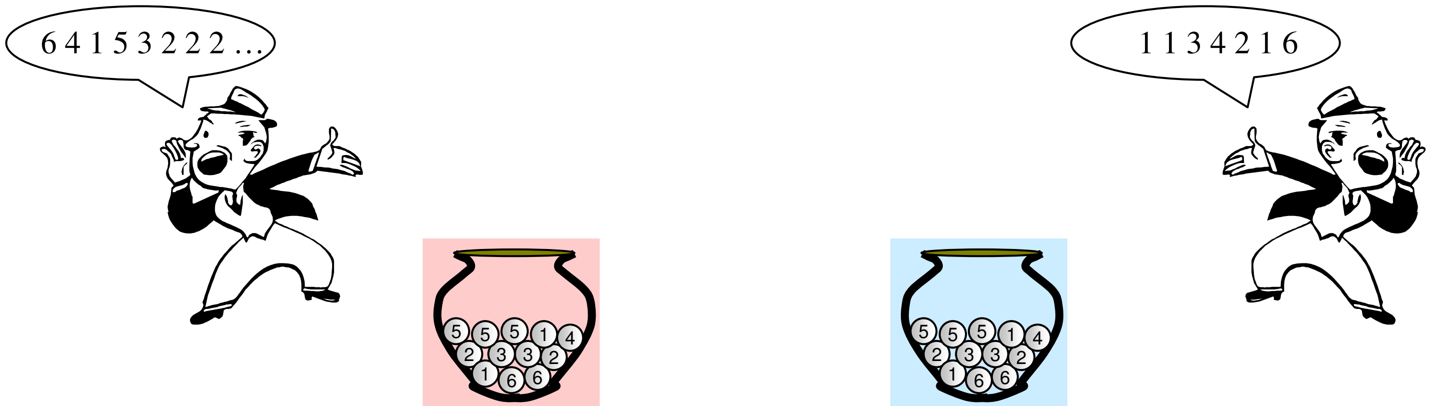
- Replacing the caller with a picker
  - Who picks balls from Urns
- Replacing the Dice with an Urn
  - Has 6 types of balls, marked “1”, “2”, “3”, “4”, “5”, “6”
  - The probability of randomly drawing a ball marked “6” = P(6 | urn)
- Picker draws a ball from the urn, calls out the number and replaces the ball in the urn
- Exactly the same model as the dice
- Problem: From the sequence of numbers called by the picker, determine the *fraction* of balls in the urns that are marked with each number.

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

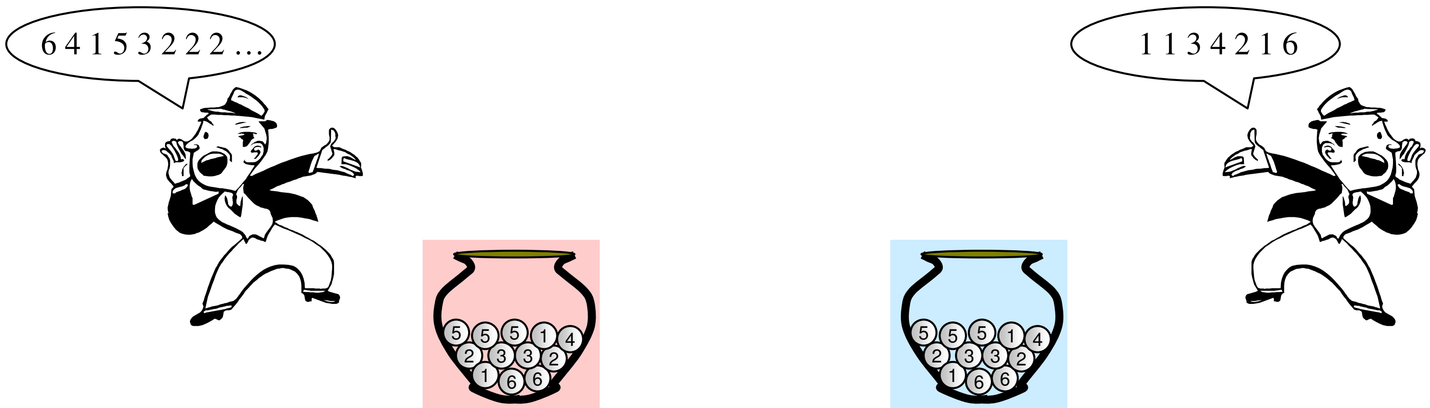
7.31 10.69

Two pickers

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

- Note: P(X|red) and P(X|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

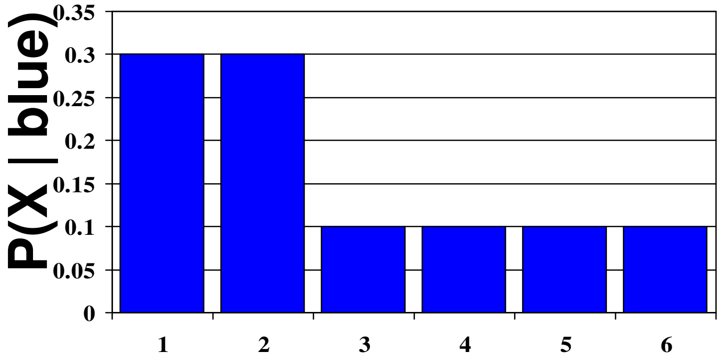
Two pickers

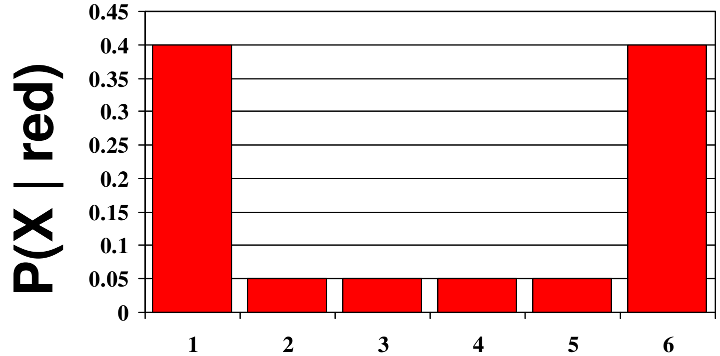


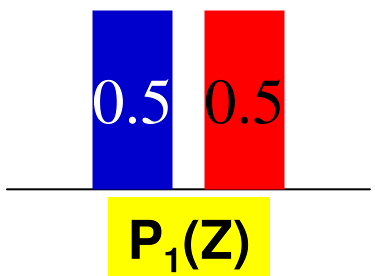
- Probability of drawing a number X for the first picker:
  - $P_1(X) = P_1(red)*P(X|red) + P_1(blue)*P(X|blue)$
- Probability of drawing X for the second picker
  - $P_2(X) = P_2(red)*P(X|red) + P_2(blue)*P(X|blue)$

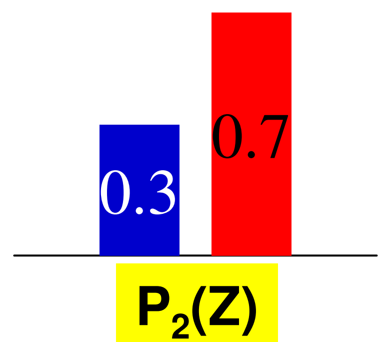
- Problem: Given the set of numbers called out by both pickers estimate
  - P_1(color) and P_2(color) for both colors
  - P(X | red) and P(X | blue) for all values of X

For the First Picker









With TWO pickers: The first picker

- Picker 1 calls: 6,4,5,1,2,3,4,5,2,2,1,4,3,4,6,2,1,6
- The table to the right is computed as before
  - Each instance of a number called is "split" between the two urns
  - The fraction of the instance going to any urn is the a posteriori probability of the urn, given the number called

$$P(color \mid observation) = \frac{P(observation \mid color) P_1(color)}{\sum_{color'} P(observation \mid color') P_1(color')}$$

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

With TWO pickers: The SECOND picker

- Picker 2 calls:
  4, 4, 3, 2, 1, 6, 5
  - Note: The number of observations is different from that for picker 1
  - In general, the *number* of observations for the two need not be the same

- We get the table to the right for the calls by picker 2
- The table is computed exactly as we computed the table for the first picker

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

$$P(color \mid observation) = \frac{P(observation \mid color) P_2(color)}{\sum_{color'} P(observation \mid color') P_2(color')}$$

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

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: 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

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: 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

- 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:

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

  - Update probability of numbers for the pots:

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

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

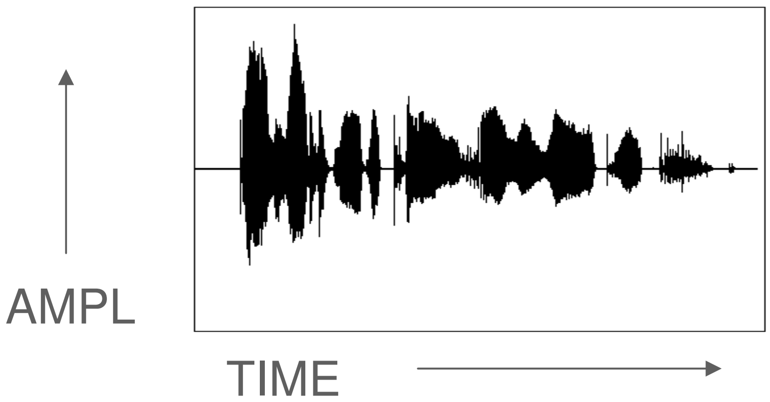
$$P_k(Z) = \frac{\sum_X N_{k,X} P(Z \mid X,k)}{\sum_{Z'} \sum_X N_{k,X} P(Z' \mid X,k)}$$

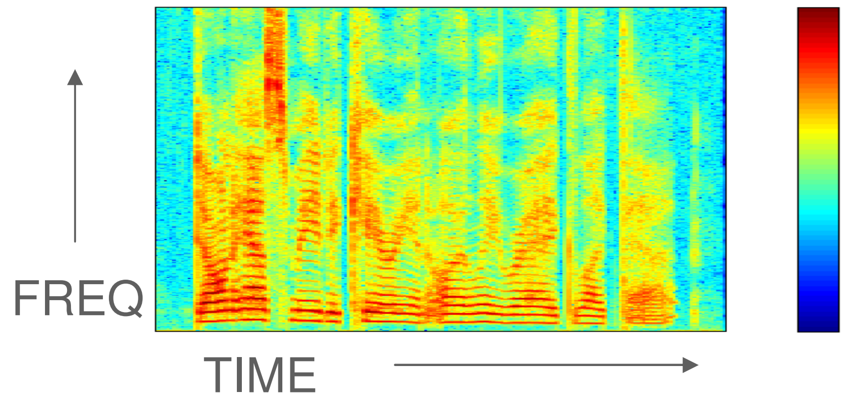
Signal Separation with the Urn model

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

- We shall see..

The representation



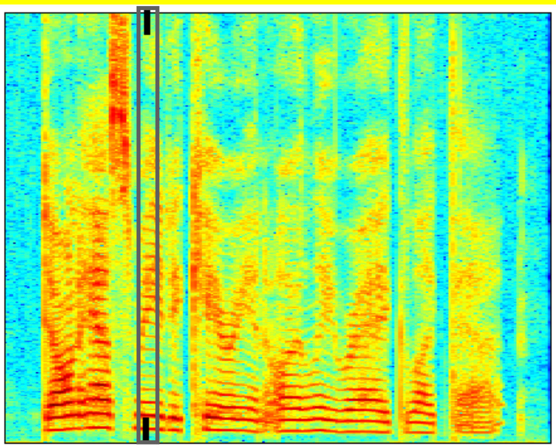


- We represent signals spectrographically
  - Sequence of magnitude spectral vectors estimated from (overlapping) segments of signal
  - Computed using the short-time Fourier transform
  - Note: Only retaining the magnitude of the STFT for our operations
  - We will, however need the phase later for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

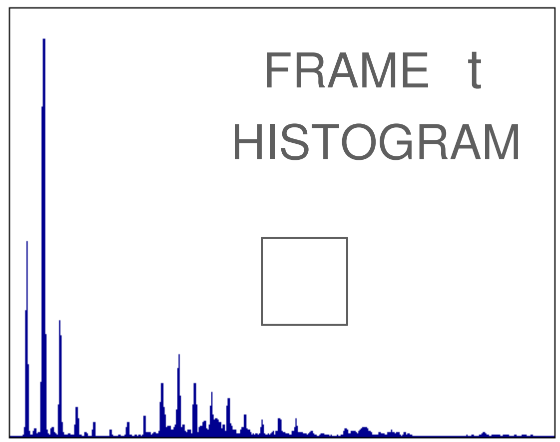
A Multinomial Model for Spectra

- A 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



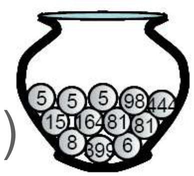
FRAME t



f

Power spectrum of frame t

$P_t(f)$

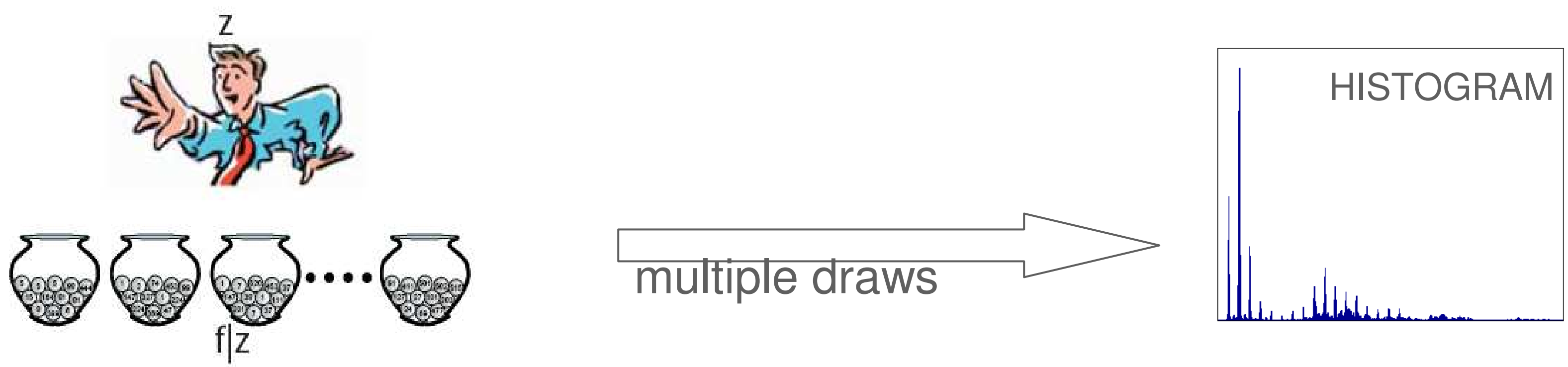


The balls are marked with discrete frequency indices from the DFT

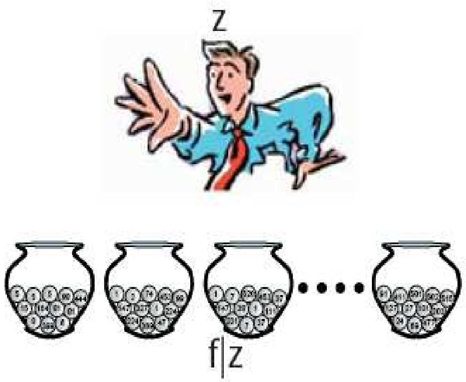
Probability distribution underlying the t-th spectral vector

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

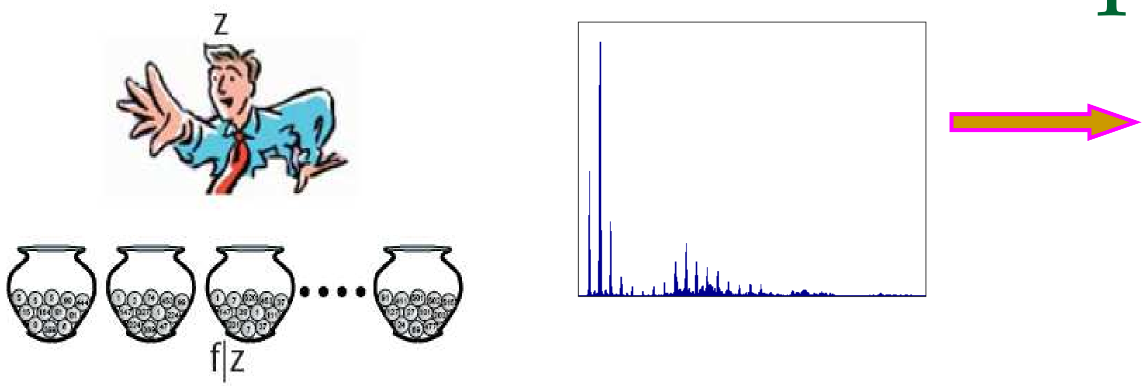


The Picker Generates a Spectrogram



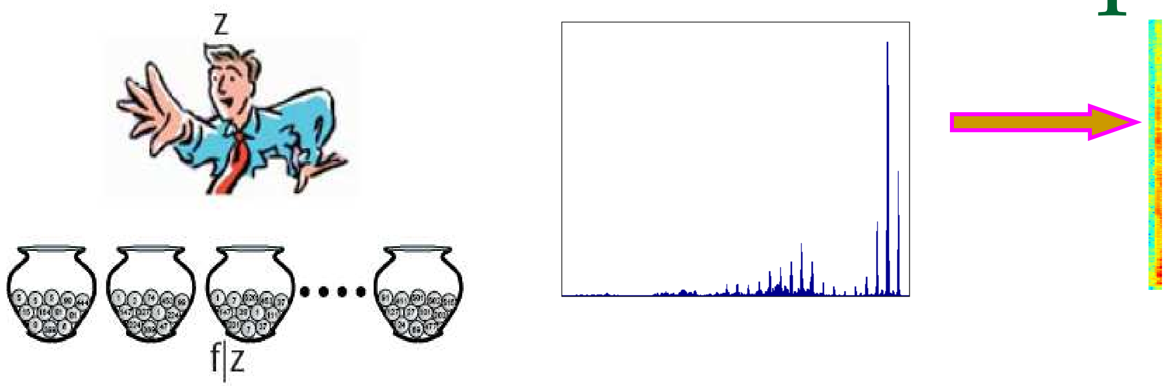
- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to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

The Picker Generates a Spectrogram



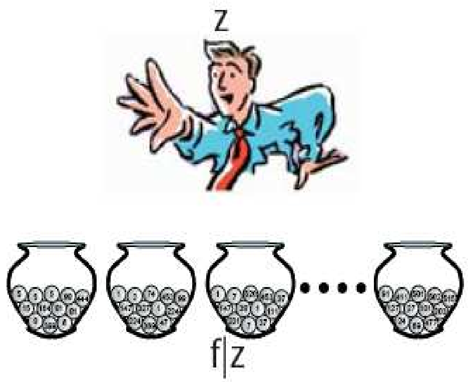
- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to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

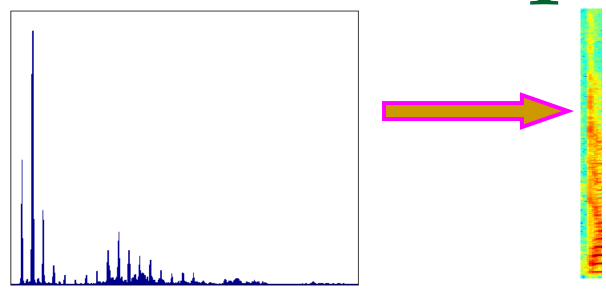
The Picker Generates a Spectrogram



- 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

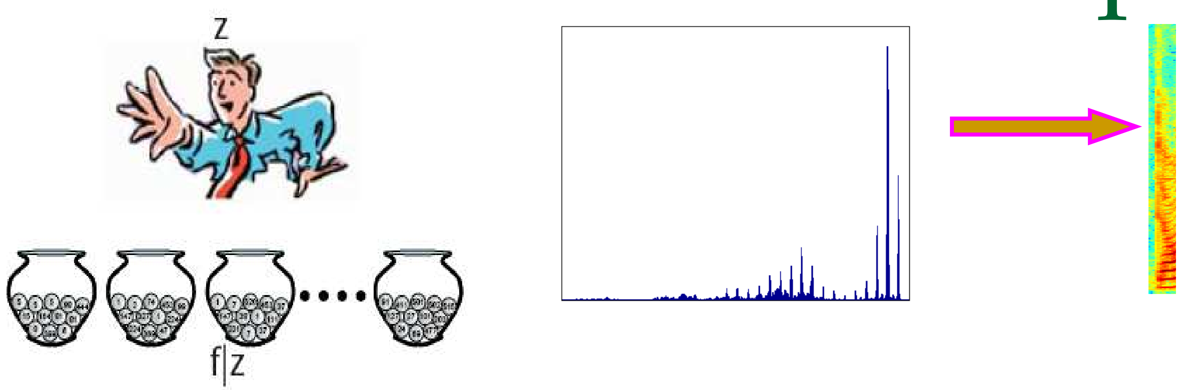
The Picker Generates a Spectrogram





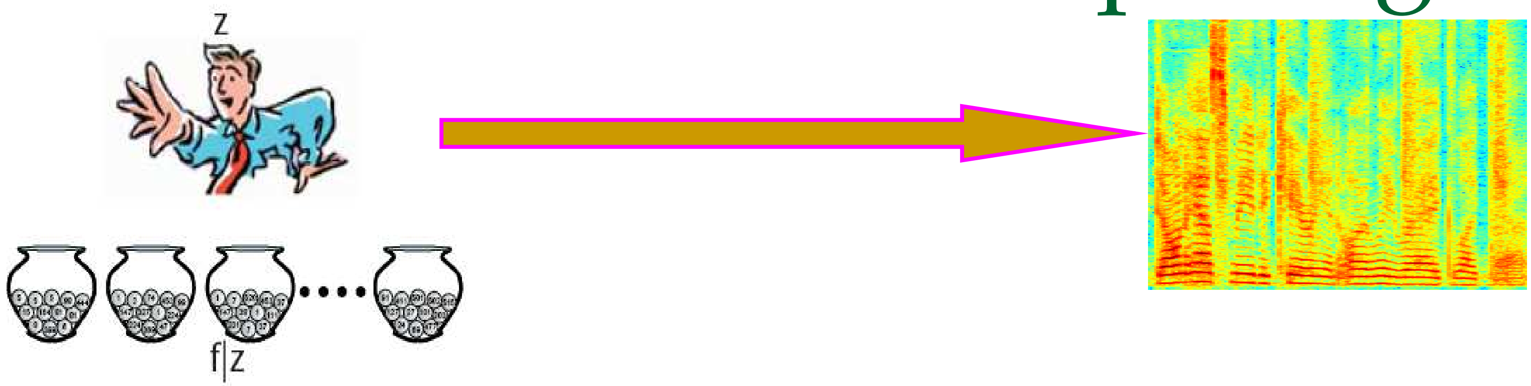
- The picker has a fixed set of Urns
  - Each urn has a different probability distribution over *f*
- He draws the spectrum for the first frame
  - In which he selects urns according to some probability $P_0(z)$
- Then draws the spectrum for the second frame
  - In which he selects urns according to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

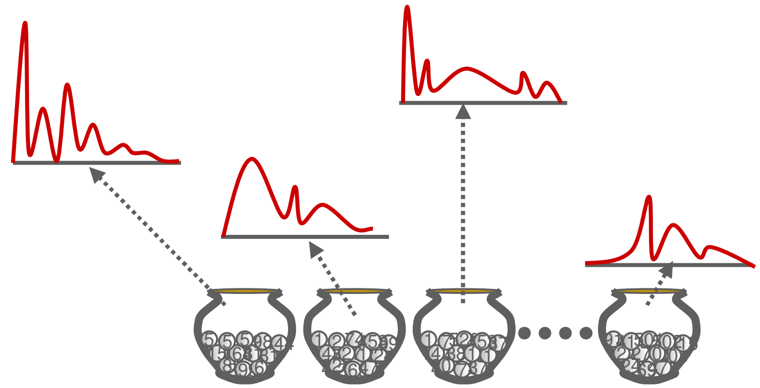
$$P_t(f) = \sum_z P_t(z) P(f \mid z)$$

Frame-specific spectral distribution ← $P_t(f)$

$P(f \mid z)$ → SOURCE specific bases

$P_t(z)$ → Frame(time) specific mixture weight

Spectral View of *Component* Multinomials

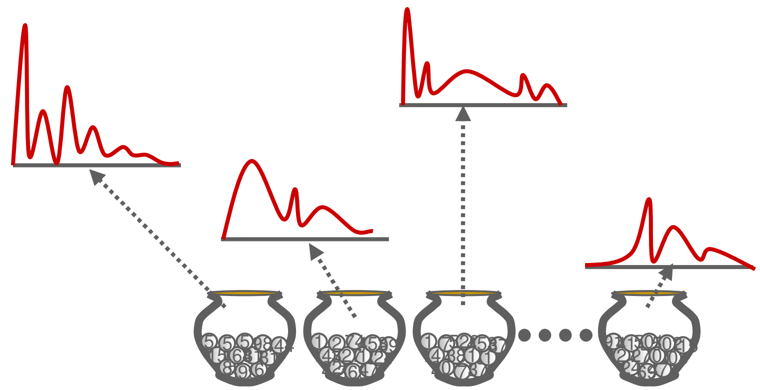


- Each component multinomial (urn) is actually a normalized histogram over frequencies *P*(*f* |z)
  - I.e. a spectrum

- 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

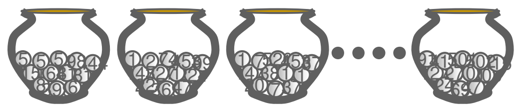


- By “learning” the mixture multinomial model for any sound source we “discover” these latent spectral structures for the source

- 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

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

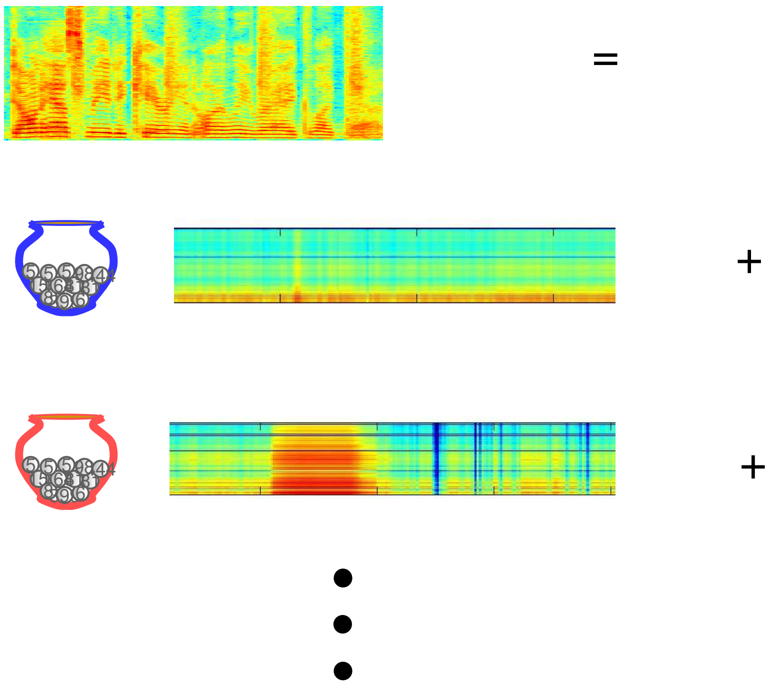
  - Compute mixture weight of z^{th} urn

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

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

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

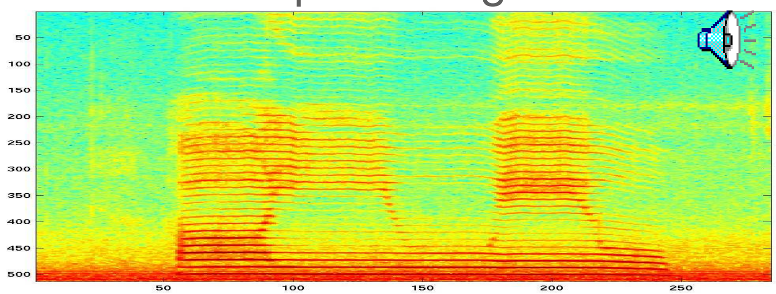
How the bases compose the signal



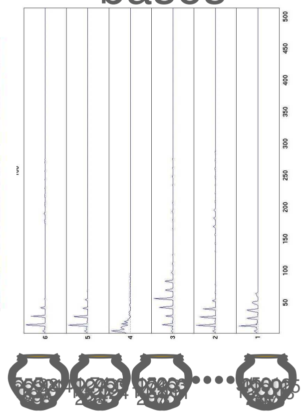
- The overall signal simply the sum of the contributions of each of the urns to the signal
  - Each urn contributes a different amount to each frame
- The contribution of the z-th urn to the t-th frame is given by $P(f|z)P_t(z)S_t$
  - $S_t = \sum_f S_t(f)$

Learning Structures

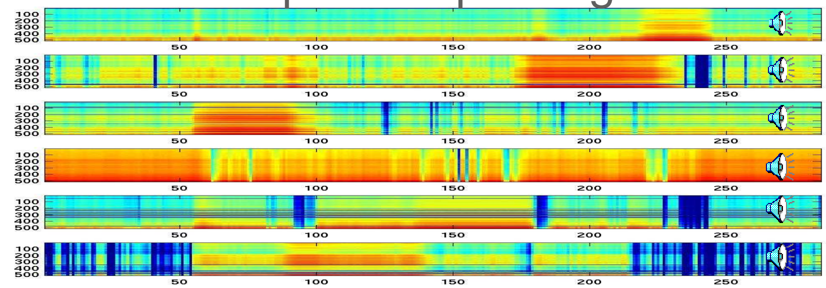
Speech Signal



bases

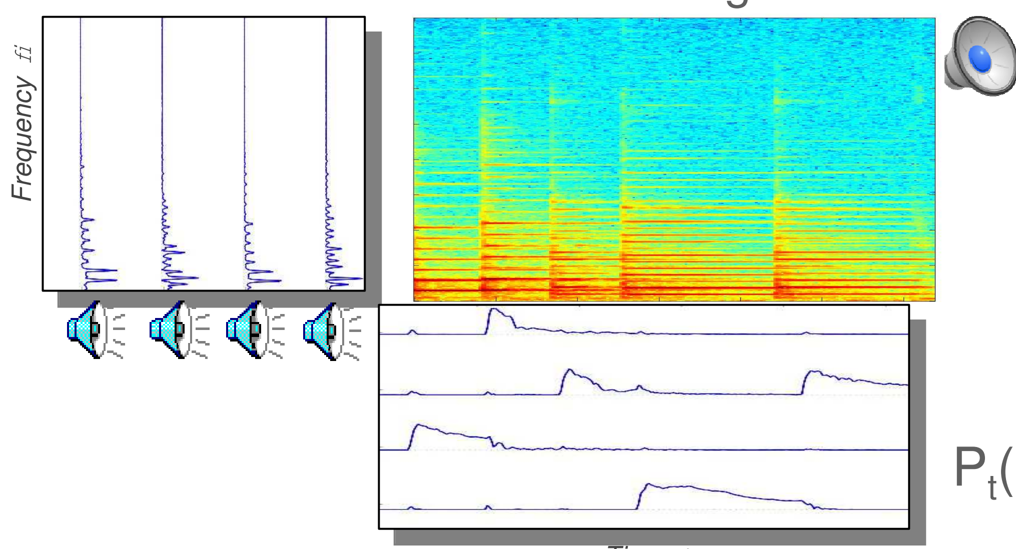


Basis-specific spectrograms



P(f|z)

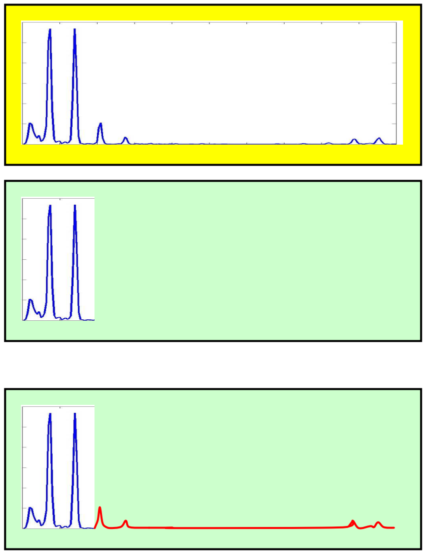
From Bach's Fugue in Gm



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**

Prediction

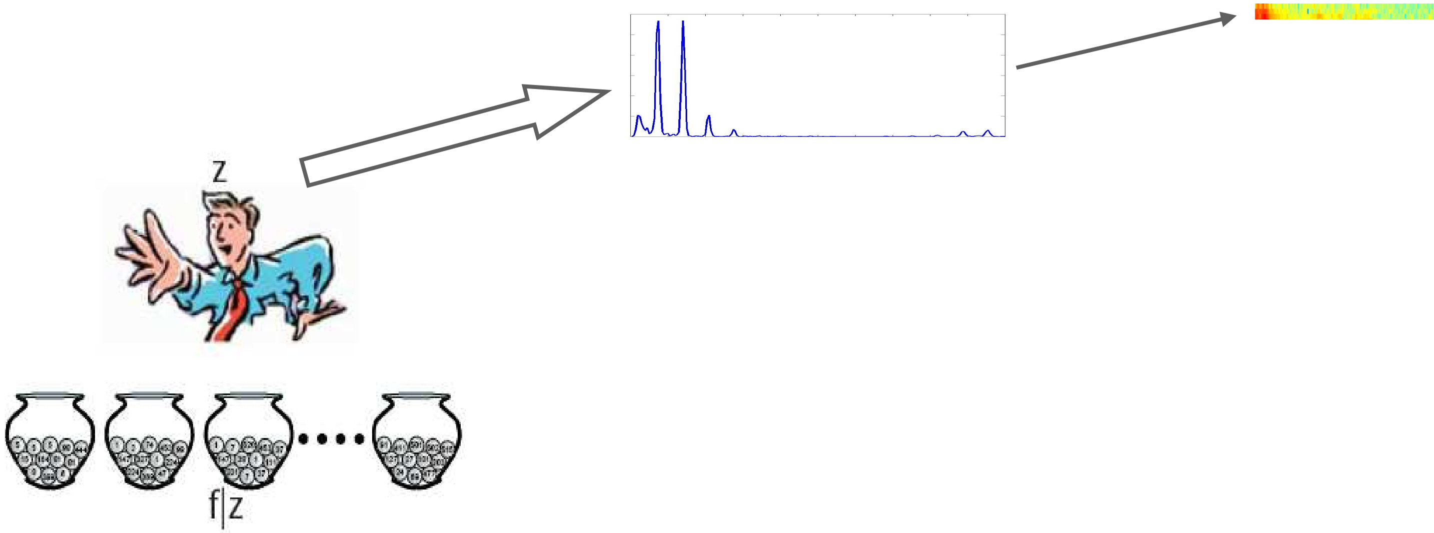


- The full basis is known
- The presence of the basis is identified from the observation of a part of it
- The obscured remaining spectral pattern can be guessed

- **Bandwidth Expansion**
  - 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

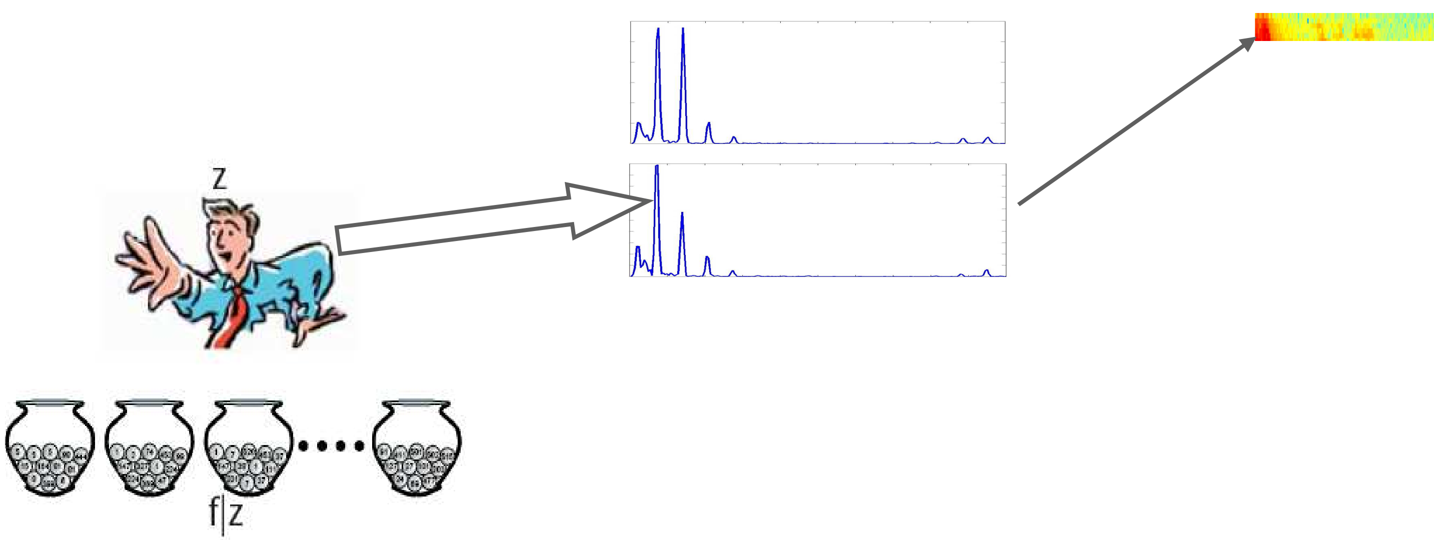
Bandwidth Expansion

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



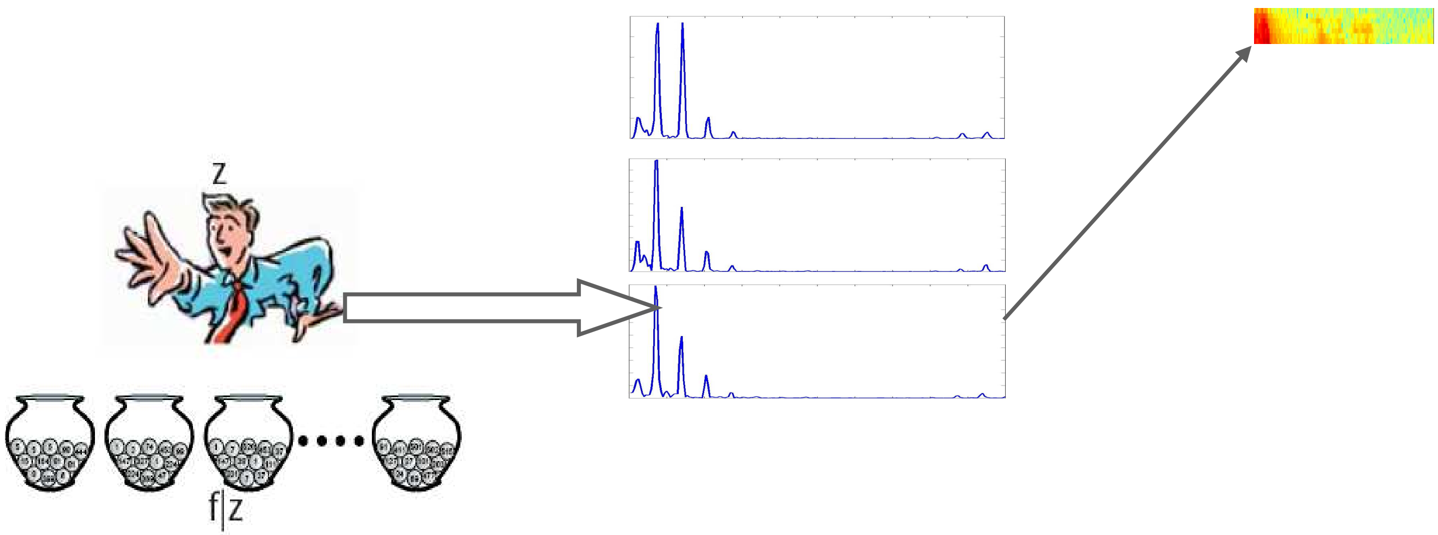
Bandwidth Expansion

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



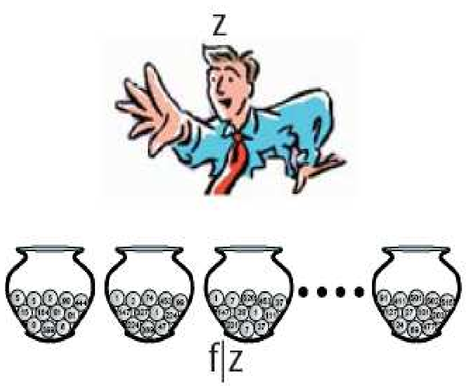
Bandwidth Expansion

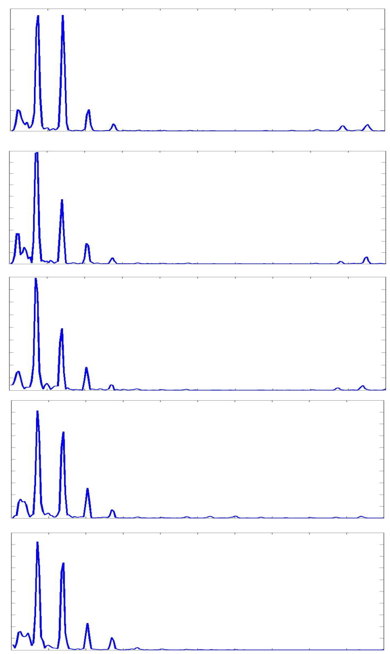
- The picker has drawn the histograms for every frame in the signal

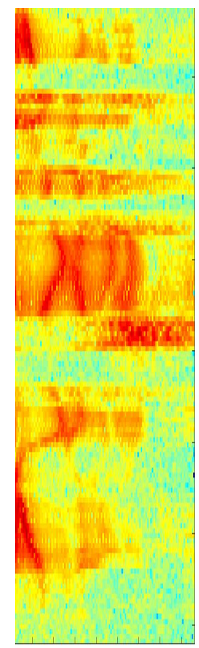


Bandwidth Expansion

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

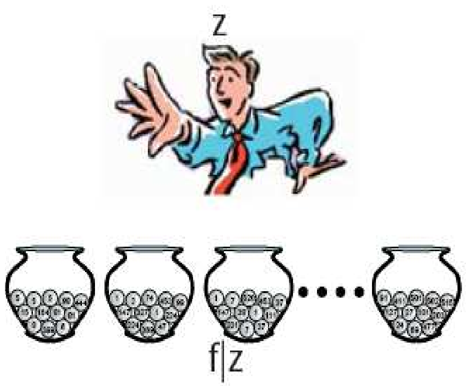


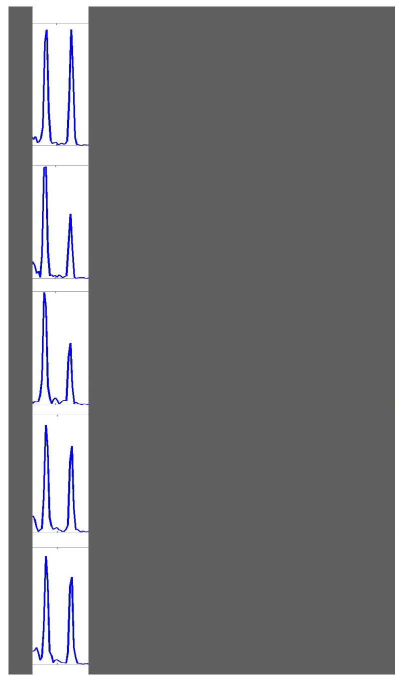


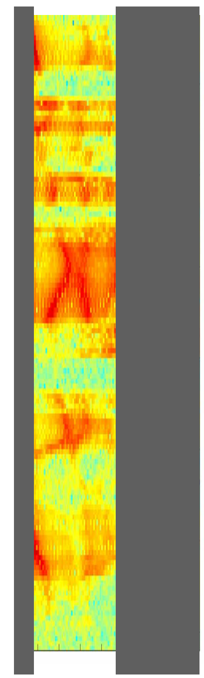


Bandwidth Expansion

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

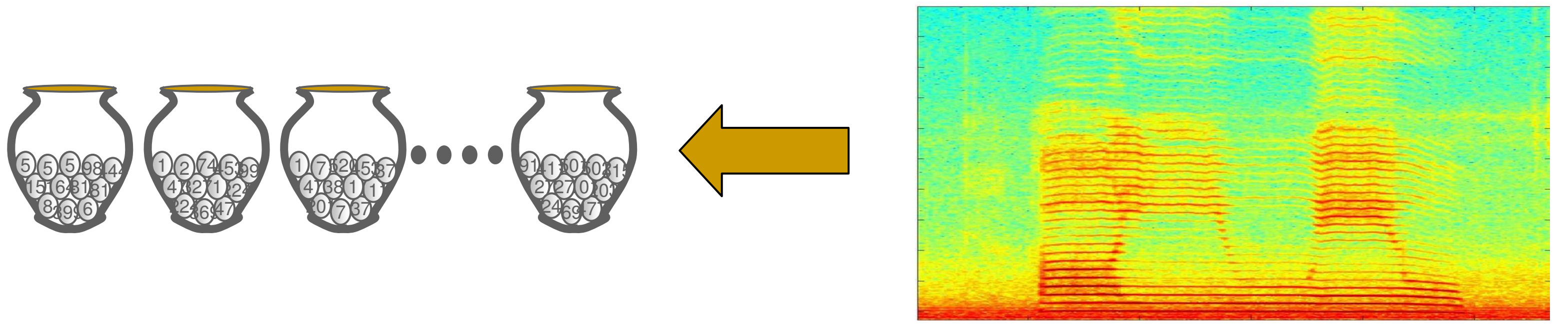






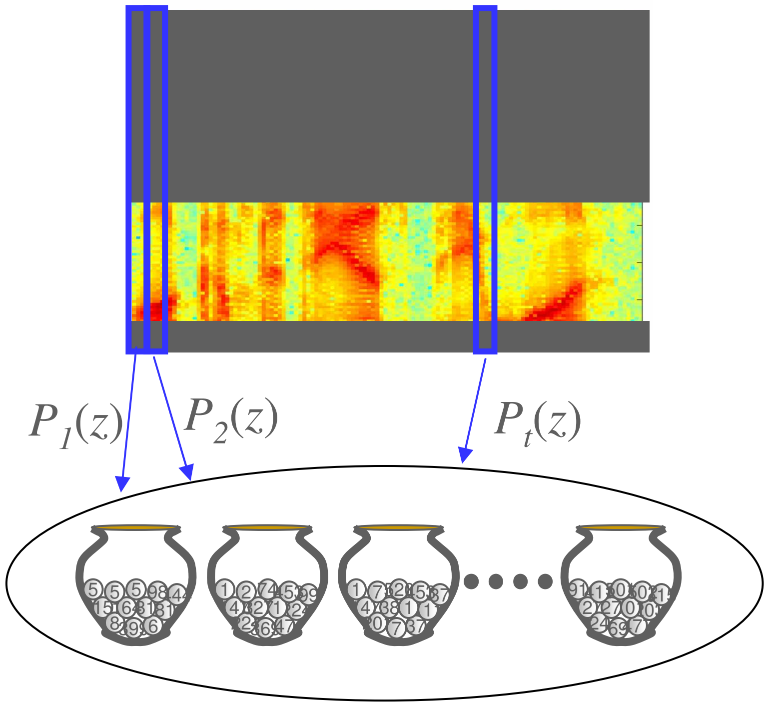
- However, we are only able to observe the number of draws of some frequencies and not the others
- We must estimate the number of draws of the unseen frequencies

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

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*

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

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

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

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

Step 3 and Step 4

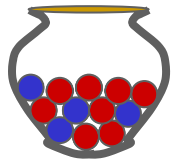
- Compose the complete probability distribution for each frame, using the mixture weights estimated in Step 2

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

- 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

Estimating unobserved frequencies

- Expected value of the number of draws:

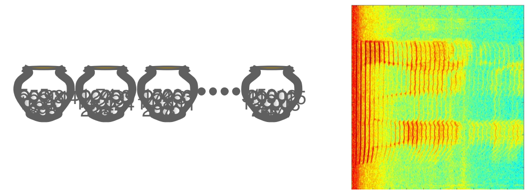
$$\hat{N}_t = \frac{\sum_{f \in (\text{observed frequencies})} S_t(f)}{\sum_{f \in (\text{observed frequencies})} P_t(f)}$$

- Estimated spectrum in unobserved frequencies

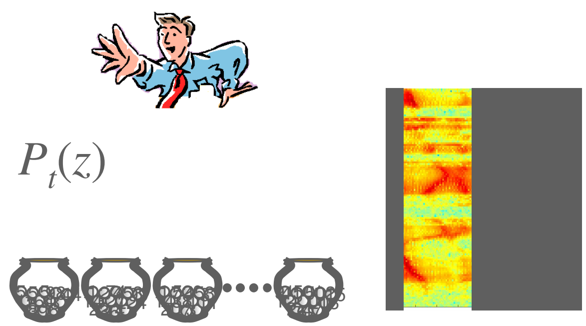
$$\hat{S}_t(f) = N_t P_t(f)$$

Overall Solution

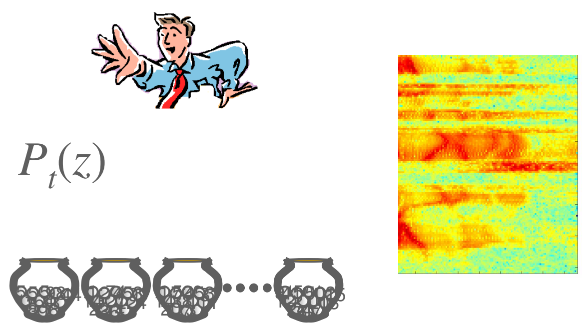
- Learn the “urns” for the signal source from broadband training data



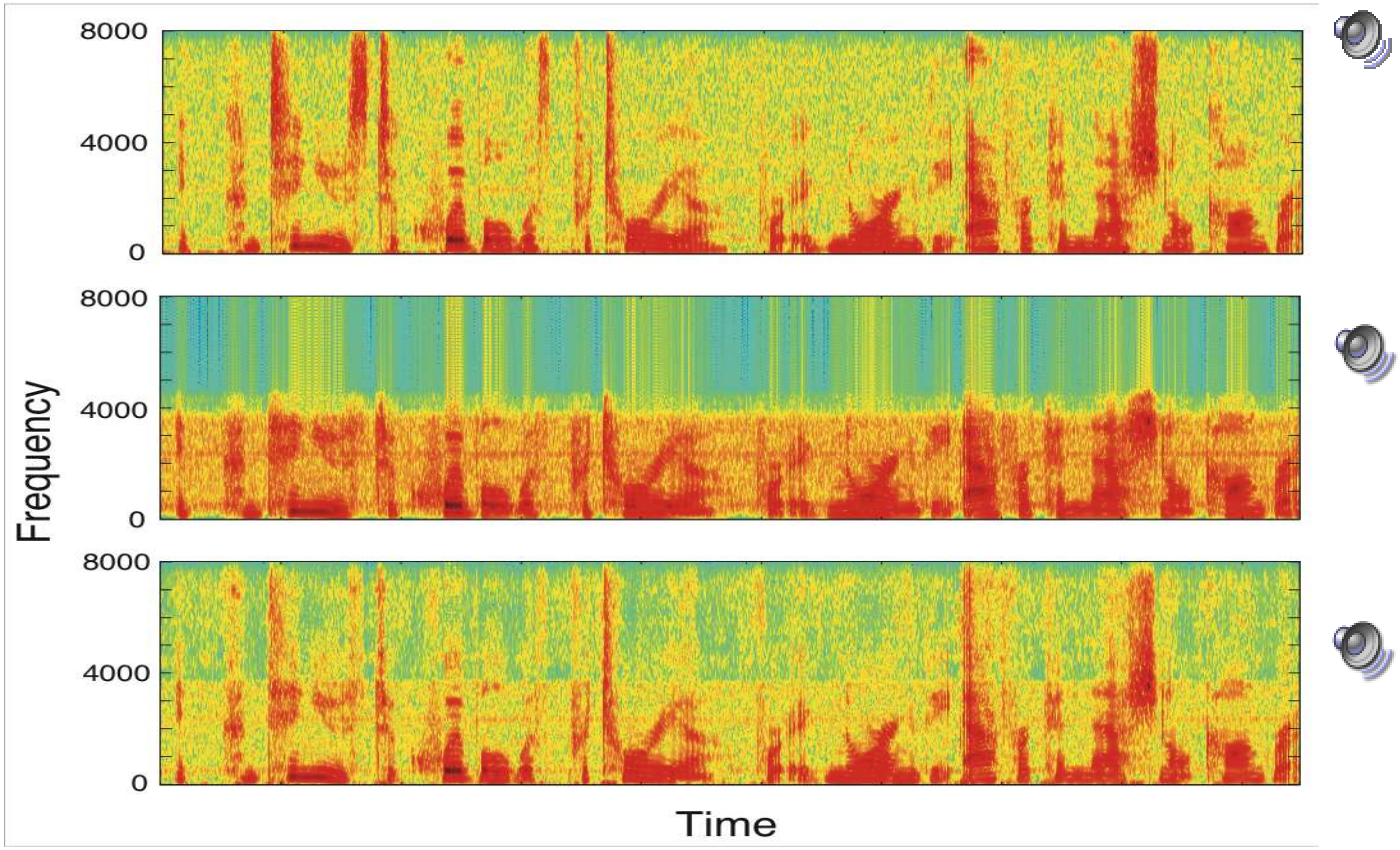
- For each frame of the reduced bandwidth test utterance, find mixture weights for the urns
  - Ignore (marginalize) the unseen frequencies



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



Some Results



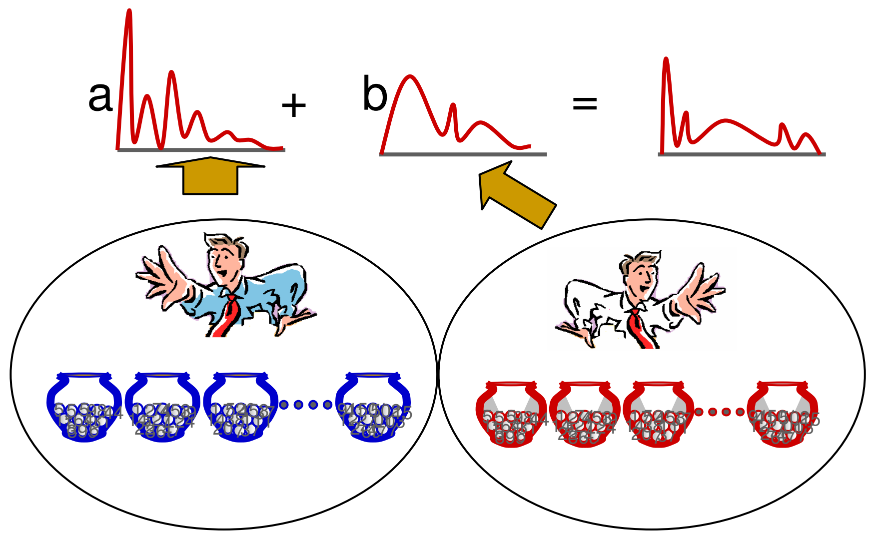
- Reasonable reconstructions are achieved
- The reconstruction is speaker specific however
  (since the urns represent spectral structures for the speaker)

Signal Separation from Monaural Recordings

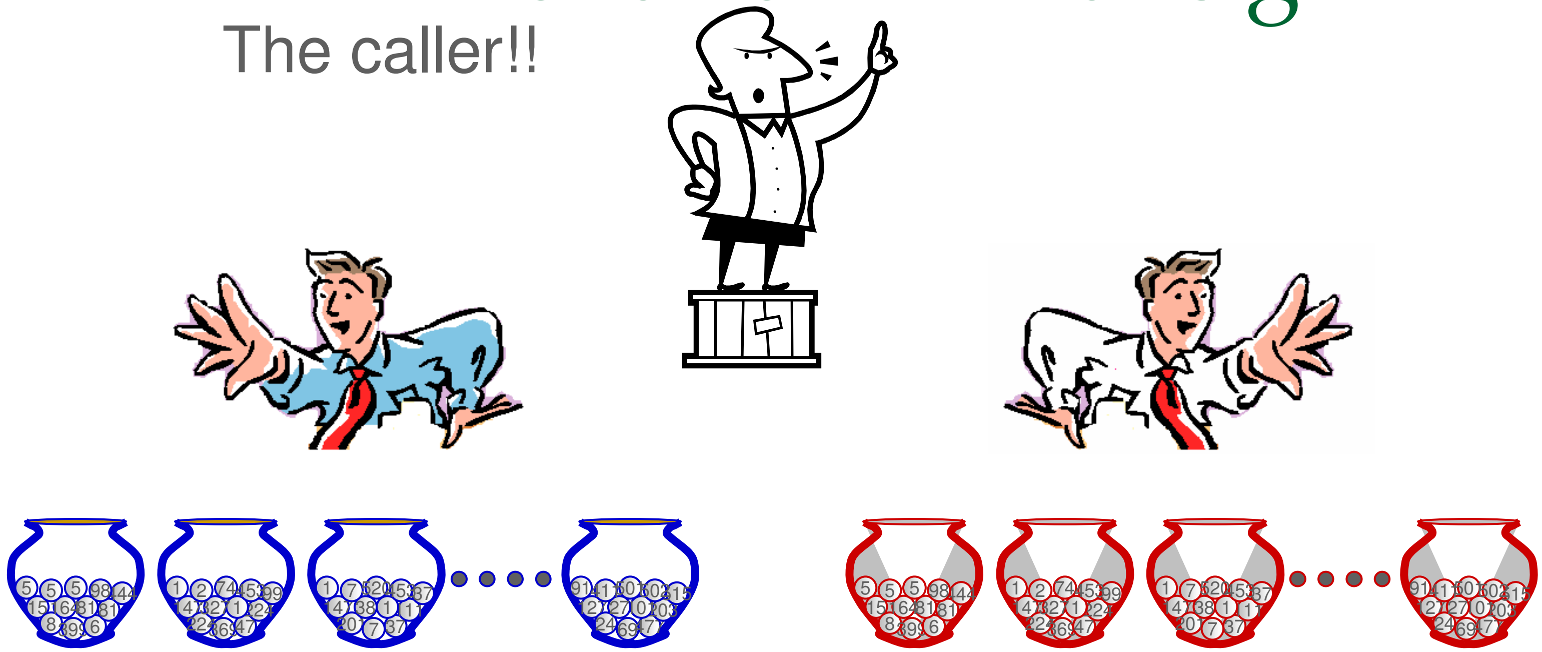
- The problem:
  - Multiple sources are producing sound simultaneously
  - The combined signals are recorded over a single microphone
  - The goal is to selectively separate out the signal for a target source in the mixture
    - Or at least to enhance the signals from a selected source

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



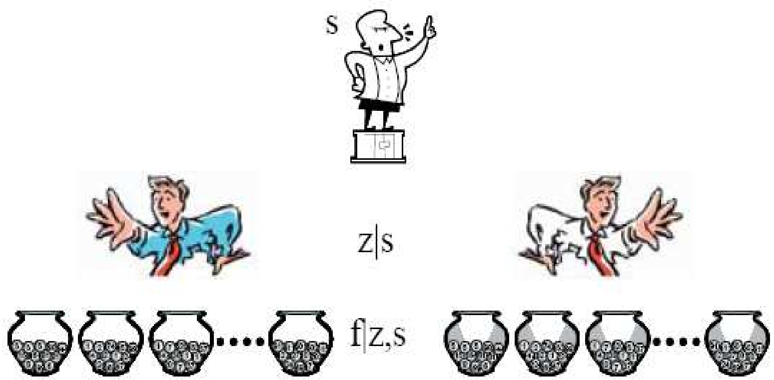
Ball-and-urn model for a mixed signal



- 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

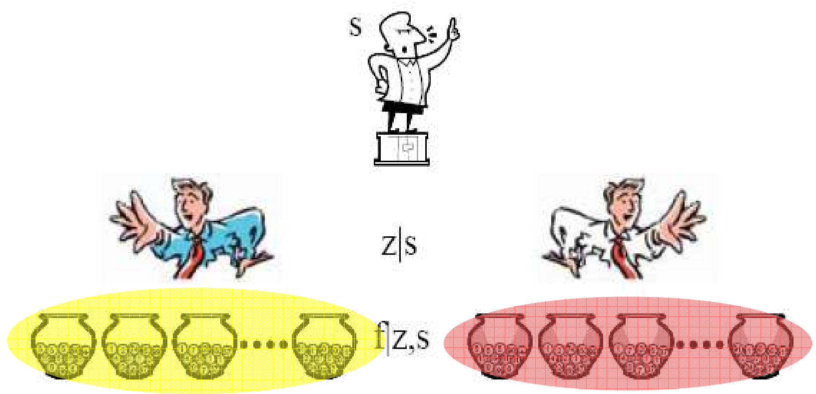


$$P_t(f) = P_t(s_1)P_t(f \mid s_1) + P_t(s_2)P_t(f \mid s_2)$$

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

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

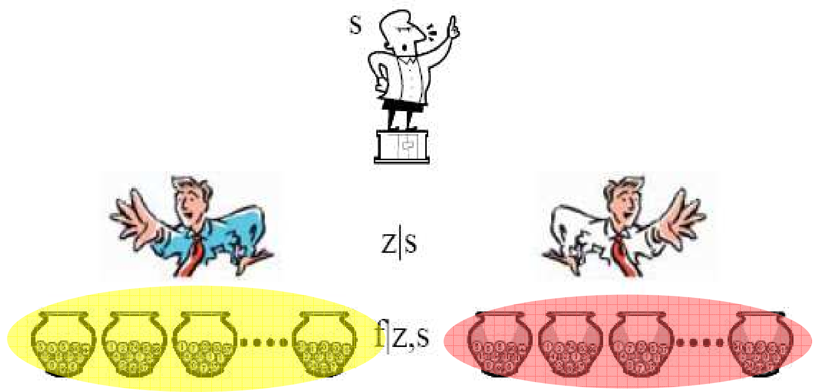


$$P_t(f) = P_t(s_1)P_t(f \mid s_1) + P_t(s_2)P_t(f \mid s_2)$$

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

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 zth 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 zth 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

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

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

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

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

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

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

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

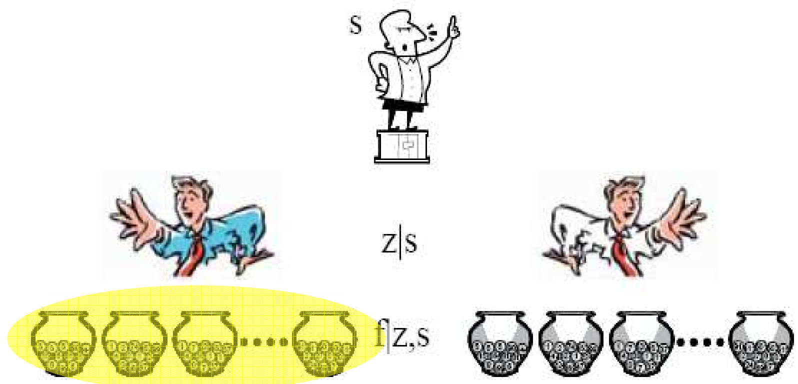
Separating the Sources

- 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

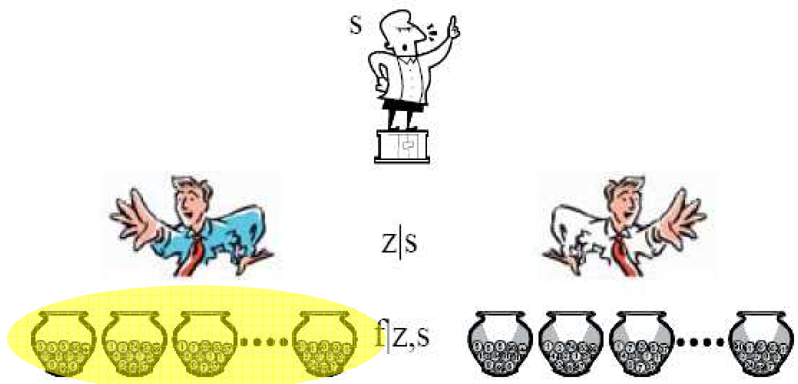


$$P_t(f) = P_t(s_1)P_t(f \mid s_1) + P_t(s_2)P_t(f \mid s_2)$$

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

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

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

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

  - Compute unknown bases

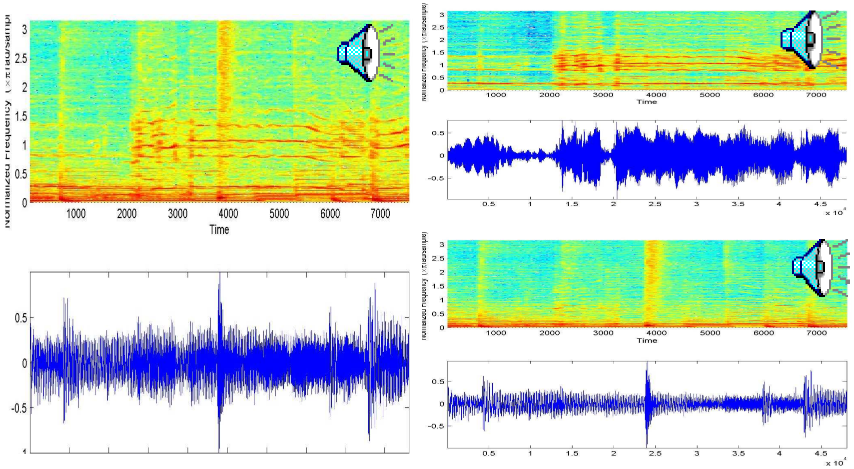
$$P(f \mid z,s) = \frac{\sum_t P_t(s,z \mid f)S_t(f)}{\sum_{f'} \sum_t P_t(s,z \mid f')S_t(f')}$$

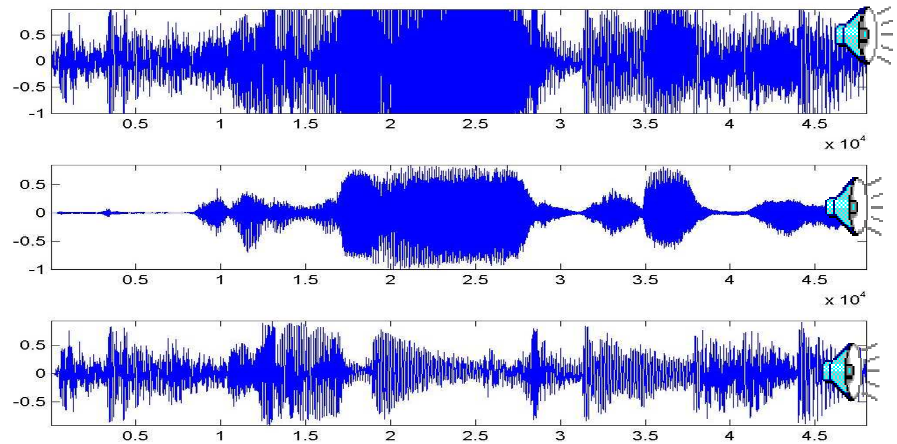
Partial information: bases for one source unknown

- 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





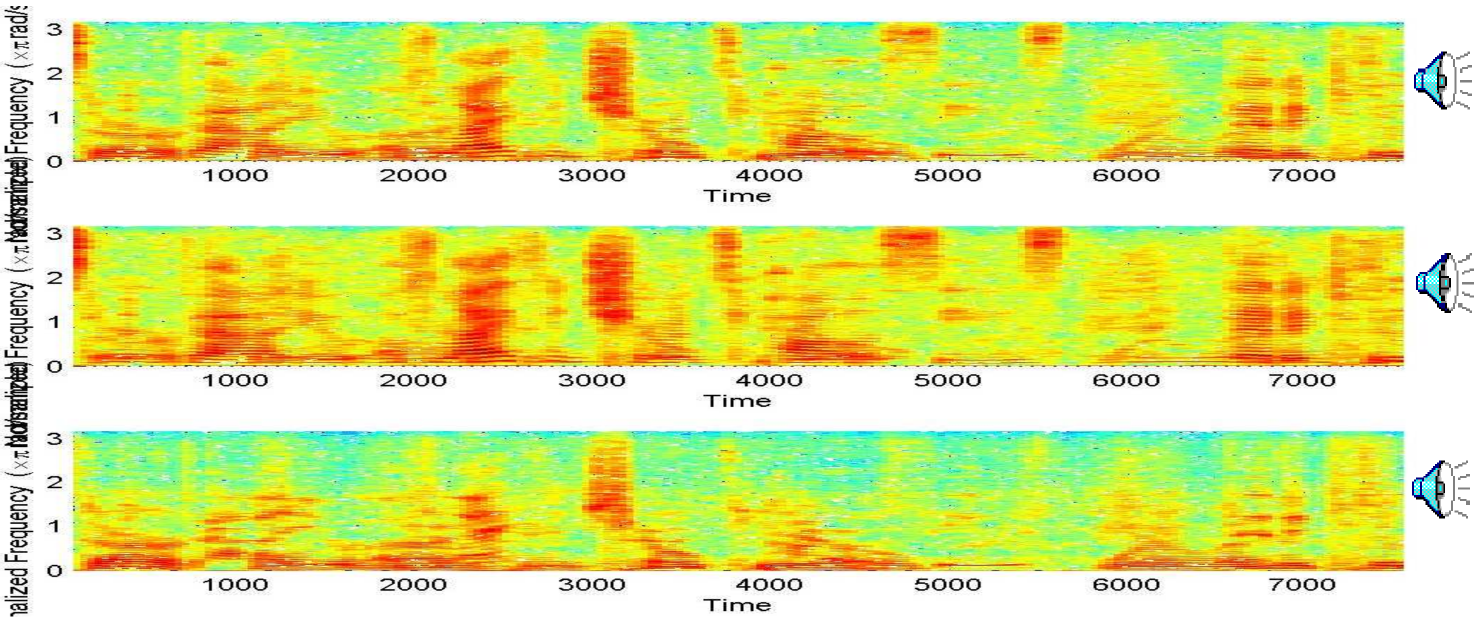
- “Raise my rent” by David Gilmour
- Background music “bases” learnt from 5-seconds of music-only segments within the song
- Lead guitar “bases” bases learnt from the rest of the song

- 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

Can it be improved?

- Yes!
- 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..

More on the topic

- Sparse overcomplete representations
- Nearest-neighbor representations
- Convolutive basis decompositions
- Transform invariance
- Etc..