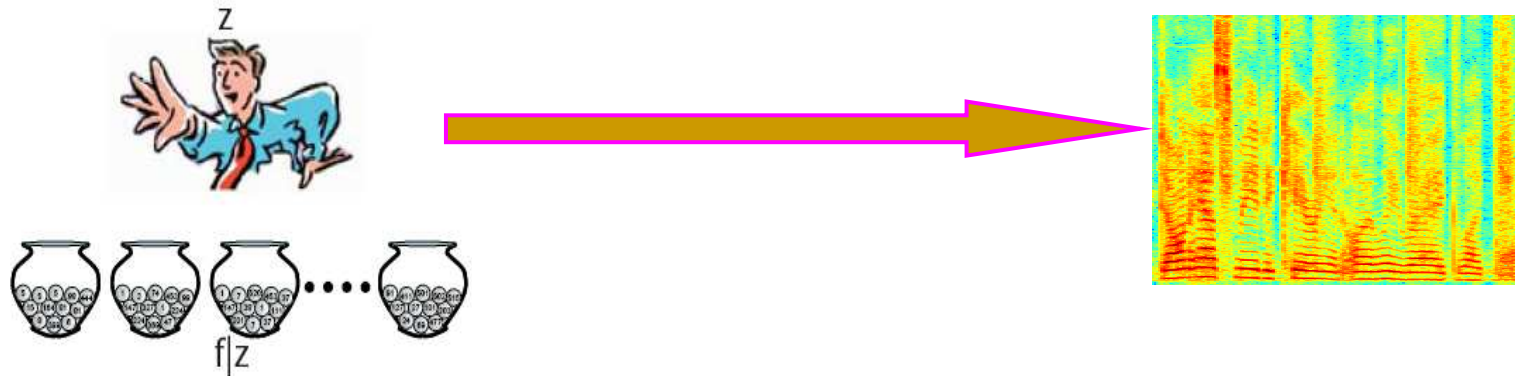
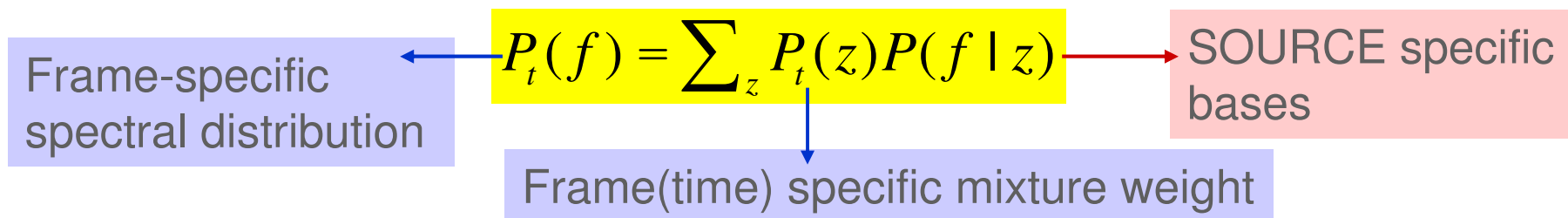

Sparse Overcomplete, Shift- and Transform-Invariant Representations

Class 15. 14 Oct 2009

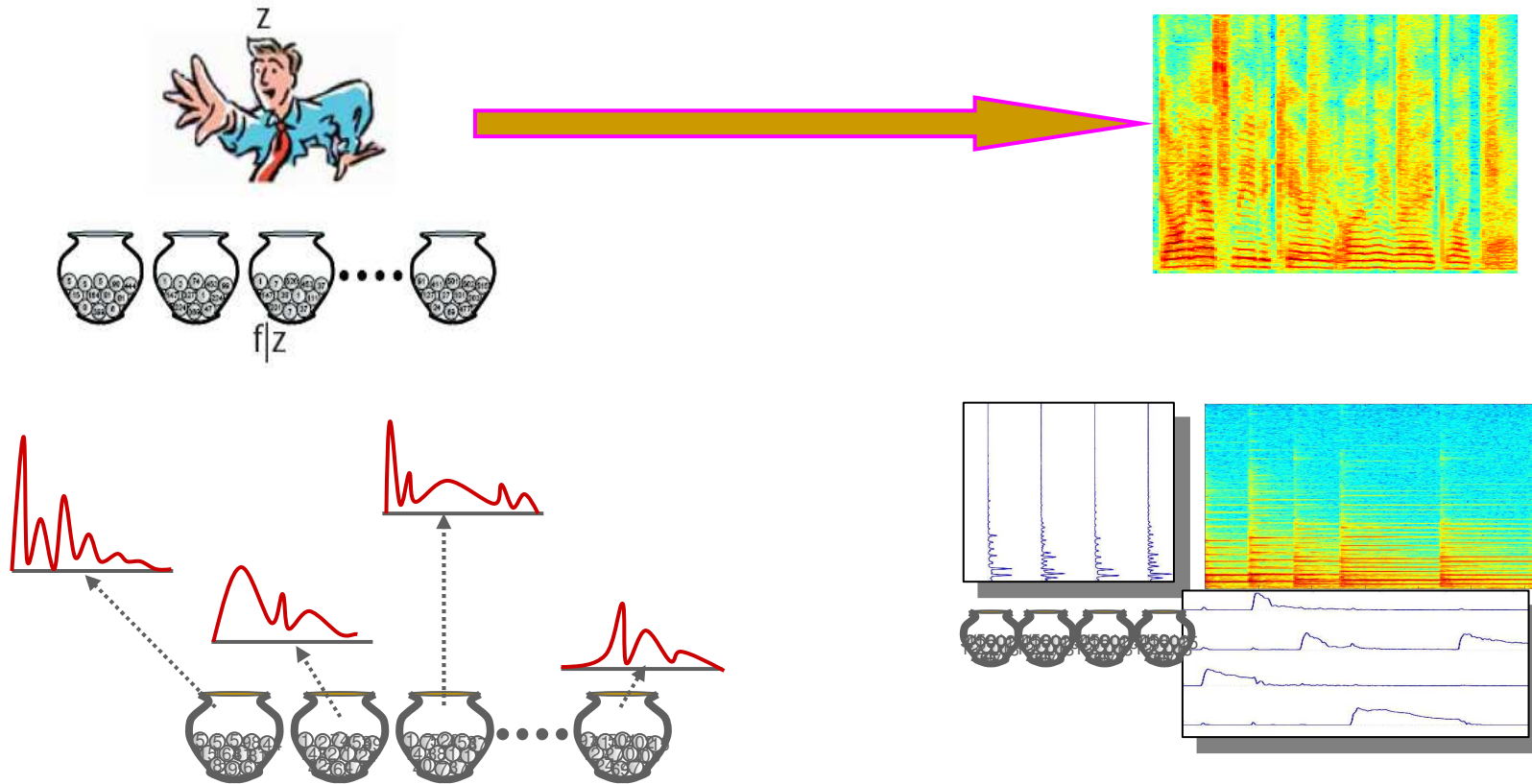
Recap: Mixture-multinomial model



- n The basic model: Each frame in the magnitude spectrogram is a histogram drawn from a mixture of multinomial (urns)
- q The probability distribution used to draw the spectrum for the t-th frame is:

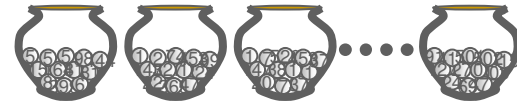
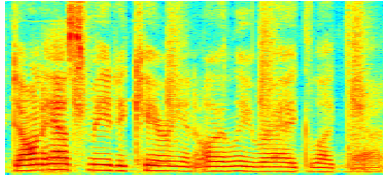


Recap: Mixture-multinomial model



- n The individual multinomials represent the “spectral bases” that compose all signals generated by the source
 - q E.g., they may be the notes for an instrument
 - q More generally, they may not have such semantic interpretation

Recap: Learning Bases



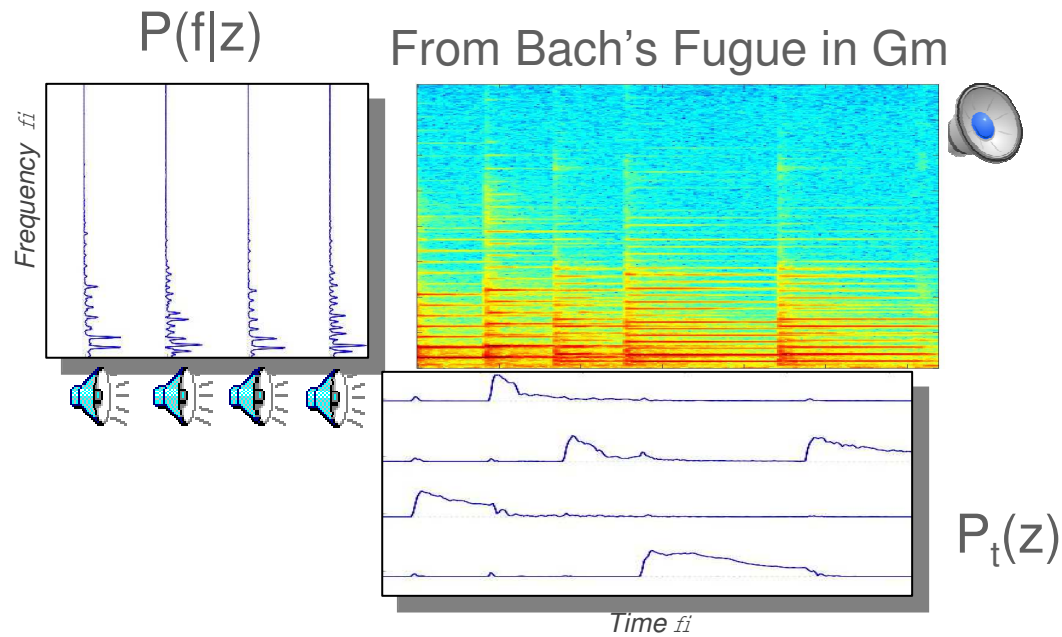
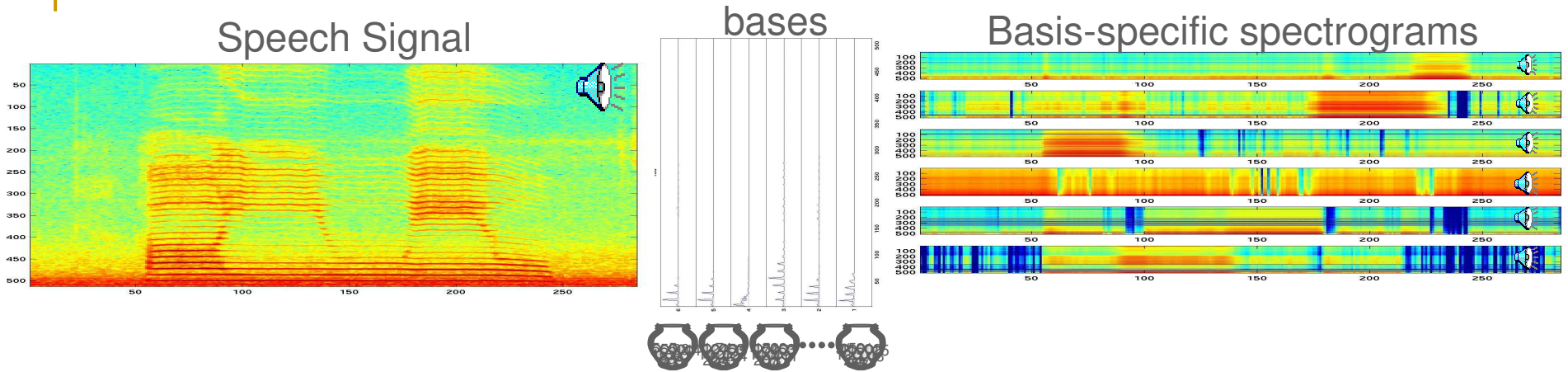
- n Learn bases from example spectrograms
- n Initialize bases ($P(f|z)$) for all z , for all f
- n For each frame, initialize $P_t(z)$
- n Iterate

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

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

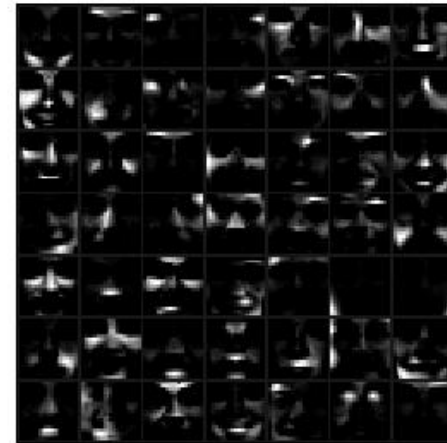
$$P(f|z) = \frac{\sum_t P_t(z|f)S_t(f)}{\sum_{f'} \sum_t P_t(z|f')S_t(f')}$$

Bases represent meaning spectral structures



How about non-speech data

19x19 images = 361 dimensional vectors



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

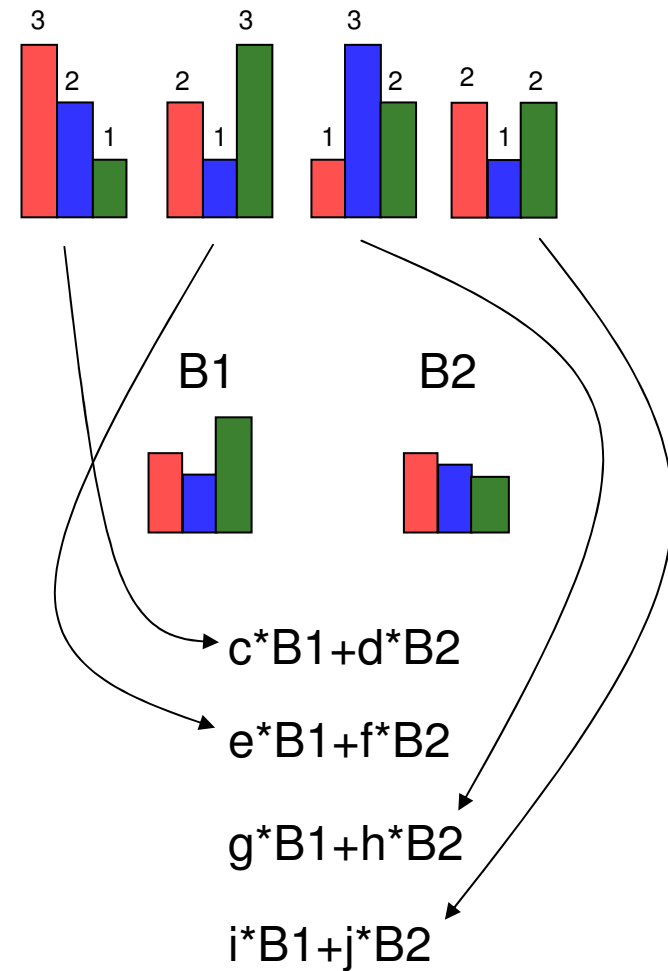
How many bases can we learn

- n The *number* of bases that must be learned is a fundamental question
 - q How do we know how many bases to learn
 - q How many bases can we actually learn computationally

- n A key computational problem in learning bases:
 - q The number of bases we can learn correctly is restricted by the dimension of the data
 - q I.e., if the spectrum has F frequencies, we cannot estimate more than $F-1$ component multinomials reliably
 - n Why?

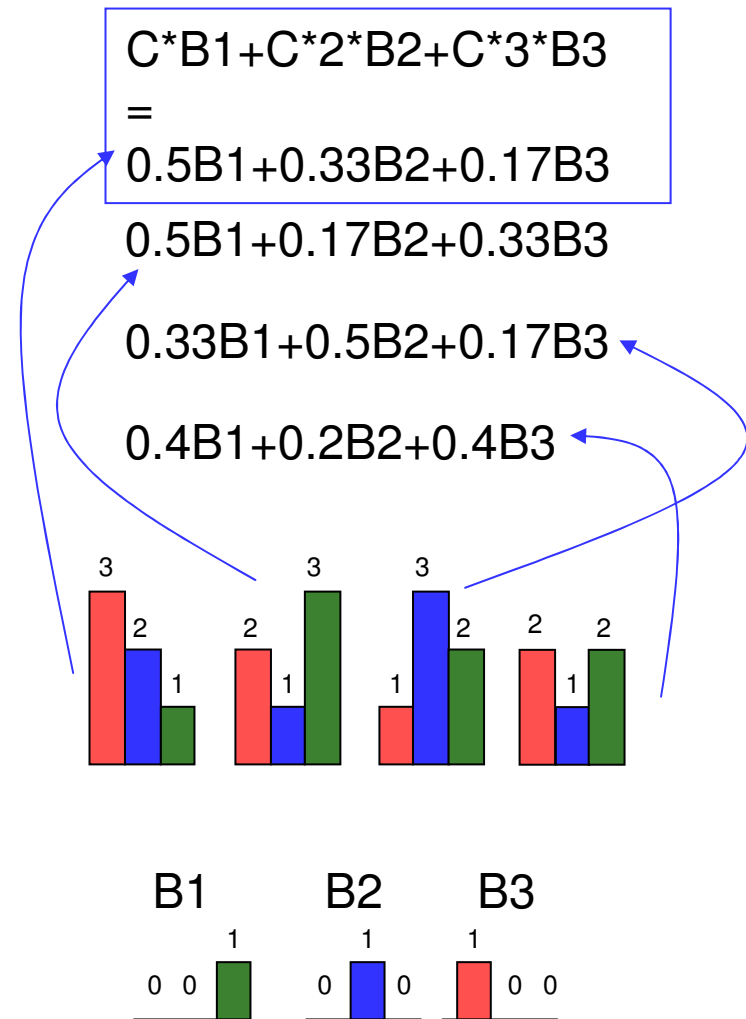
Indeterminacy in Learning Bases

- n Consider the four histograms to the right
- n All of them are mixtures of the same K component multinomials
- n For $K < 3$, a single global solution may exist
 - q I.e there may be a unique set of component multinomials that explain all the multinomials
 - n With error – model will not be perfect
- n For $K = 3$ a trivial solution exists



Indeterminacy

- n Multiple solutions for $K = 3$..
 - q We cannot *learn* a non-trivial set of “optimal” bases from the histograms
 - q The component multinomials we do learn tell us nothing about the data
- n For $K > 3$, the problem only gets worse
 - q An infinite set of solutions are possible
 - n E.g. the trivial solution plus a random basis



Indeterminacy in signal representations

n Spectra:

- q If our spectra have D frequencies (no. of unique indices in the DFT) then..
- q We cannot learn D or more meaningful component multinomials to represent them
 - n The trivial solution will give us D components, each of which has probability 1.0 for one frequency and 0 for all others
 - n This does not capture the innate spectral structures for the source

n Images: Not possible to learn more than $P-1$ meaningful component multinomials from a collection of P -pixel images

Overcomplete Representations

- n Representations where there are more bases than dimensions are called *Overcomplete*
 - q E.g. more multinomial components than dimensions
 - q More L2 bases (e.g. Eigenvectors) than dimensions
 - q More non-negative bases than dimensions
- n Overcomplete representations are difficult to compute
 - q Straight-forward computation results in indeterminate solutions
- n Overcomplete representations are required to represent the world adequately
 - q The complexity of the world is not restricted by the dimensionality of our representations!

How many bases to represent sounds/images?

- n In each case, the bases represent “typical unit structures”
 - q Notes
 - q Phonemes
 - q Facial features..
- n *To model the data well, all of these must be represented*
- n How many notes in music
 - q Several octaves
 - q Several instruments
- n The total number of notes required to represent all “typical” sounds in music are in the thousands
- n The typical sounds in speech –
 - q Many phonemes, many variations, can number in the thousands
- n Images:
 - q Millions of units that can compose an image – trees, dogs, walls, sky, etc. etc. etc...

How many can we learn

- n Typical Fourier representation of sound: 513 (or less) unique frequencies
 - q I.e. no more than 512 unique bases can be learned reliably
 - q These 512 bases must represent *everything*
 - n Including the units of music, speech, and the other sounds in the world around us
 - q Depending on what we're attempting to model

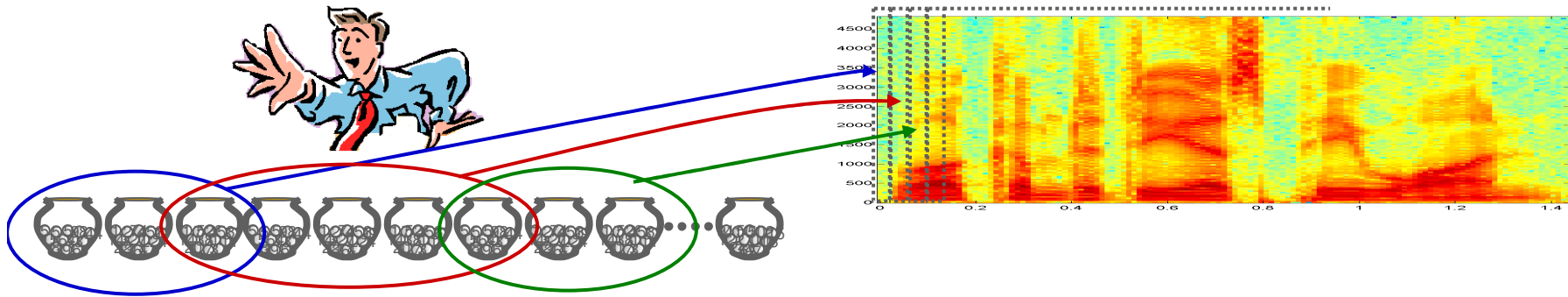
- n Typical “tiny” image: 100x100 pixels
 - q 10000 pixels
 - q I.e. no more than 9999 distinct bases can be learned reliably
 - q But the number of unique entities that can be represented in a 100x100 image is countless!

- n We need *overcomplete* representations to model these data well

Learning Overcomplete Representations

- n Learning more multinomial components than dimensions (frequencies or pixels) in the data leads to indeterminate or useless solution
- n Additional criteria must be imposed in the learning process to learn more components than dimensions
 - q Impose additional constraints that will enable us to obtain meaningful solutions
- n We will require our solutions to be *sparse*

SPARSE Decompositions



n Allow any arbitrary number of bases (urns)

q Overcomplete

n Specify that for any *specific* frame only a small number of bases may be used

q Although there are many spectral structures, any given frame only has a few of these

n In other words, the mixture weights with which the bases are combined must be sparse

q Have non-zero value for only a small number of bases

q Alternately, be of the form that only a small number of bases contribute significantly

The history of sparsity

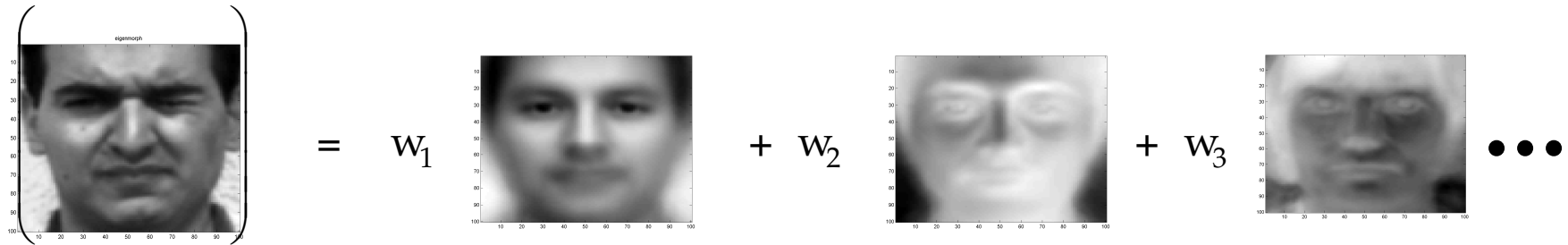
- n The search for “sparse” decompositions has a long history
 - q Even outside the scope of overcomplete representations

- n A landmark paper: Sparse Coding of Natural Images Produces Localized, Oriented, Bandpass Receptive Fields, by Olshausen and Fields
 - q *“The images we typically view, or natural scenes, constitute a minuscule fraction of the space of all possible images. It seems reasonable that the visual cortex, which has evolved and developed to effectively cope with these images, has discovered efficient coding strategies for representing their structure. Here, we explore the hypothesis that the coding strategy employed at the earliest stage of the mammalian visual cortex maximizes the sparseness of the representation. We show that a learning algorithm that attempts to find linear sparse codes for natural scenes will develop receptive fields that are localized, oriented, and bandpass, much like those in the visual system.”*
 - q Images can be described in terms of a small number of descriptors from a large set
 - n E.g. a scene is “a grapevine plus grapes plus a fox plus sky”

- n Other studies indicate that human perception may be based on sparse compositions of a large number of “icons”

- n The number of sensors (rods/cones in the eye, hair cells in the ear) is much smaller than the number of visual / auditory objects in the world around us
 - q The representation is overcomplete

Representation in L2



n Conventional Eigen Analysis:

q Compute Eigen Vectors such that $\|X - EW\|^2$ is minimized

n The columns of E are orthogonal to one another

n Eigen analysis is an “L2” decomposition

q Minimizes the L2, or Euclidean error in composition

n The maximum number of Eigen vectors = no. of dimensions D

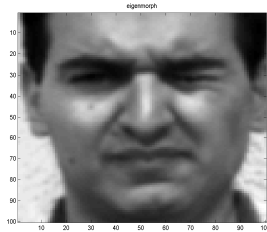
n We could use *any* set of D linearly independent vectors (e.g. a DxD matrix B); not only the Eigen vectors

q The data vector could be expressed in the same manner as above

q The only distinction will now be that unlike E, the columns of A are no longer orthogonal

q The weights with which the bases must be combined are obtained by a $\text{pinv}(B) * X$

Overcomplete representations in L2



= Linear combination of



More bases than
Pixels

n Sparse L2 representation

q Minimize $\|X - BW\|^2$

n Same as before, except the number of bases are much greater than the number of dimensions

q The bases are *no longer Eigen vectors*

n The weights w_i must now be sparse

q I.e. although the number of bases is $> D$, the number of non-zero weight terms for any data X must be less than D

n Conventional dot product / psuedoinverse-based algorithms will not give us the correct solution

q They impose no constraint on W

Sparse overcomplete representations in L2

n Problem:

- q Given an overcomplete set of bases B_1, B_2, \dots, B_N
- q Estimate the weights w_1, w_2, \dots, w_N such that
- q $X = w_1 B_1 + w_2 B_2 + \dots + w_N B_N$
 - n X is D dimensional; $D < N$
- q And the set of weights $\{w_i\}$ is sparse

n Problem formulation:

- q $\text{Argmin}_W \|X - BW\|^2 + \text{Constraint}(W)$
- q W is the set of weights in vector form
- q The “constraint” is a sparsity constraint
- q Given many equivalent unconstrained solutions for W , it forces the selection of the sparsest of these solutions

Sparse L2 Decomposition

n Problem formulation:

q $\text{Argmin}_W ||X - BW||^2 + \text{Constraint}(W)$

n The L0 constraint

q Objective to minimize = $||X - BW||^2 + |W|_0$

q Minimizes error of reconstruction AND minimizes the number of non-zero terms in W

n L0 norm $|W|_0$ = the number of non-zero terms by definition

q Computationally intractable for large basis sets

n Needs a combinatorial search

q Approximate solutions:

n COSamp

n L2 solution with flooring

n Etc.

Sparse L2 Decomposition

- n Problem formulation:

- q $\text{Argmin}_W \|X - BW\|^2 + |W|_1$

- n $|W|_1$ is the L1 norm of W

- n i.e. the sum of the magnitude of all entries in W

- n The L1 constraint

- q Minimization of L0 is computationally intractable

- q Under certain generic conditions, it is sufficient to minimize the L1 norm instead

- n “Restricted Isometry” of B

- n The optimal L1 solution will also be the optimal L0 solution

- n L1 minimization is a standard convex optimization problem

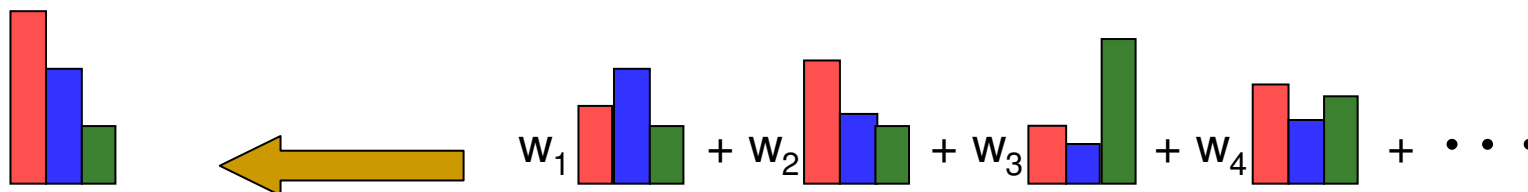
- q Downloadable code is available from Caltech (the L1 magic package):

- q <http://www.acm.caltech.edu/l1magic/>

Learning Overcomplete L2 Representations

- n We have seen how to estimate *weights* given bases
- n How about learning the optimal set of bases?
- n Sparse PCA:
 - q Learn Orthogonal Eigen-like vectors that can be combined sparsely
 - q Cannot be overcomplete
- n Random projections
- n Other techniques for learning “dictionaries” for overcomplete bases
- n Good information on Dave Donoho’s Stanford page

Sparsity and Overcompleteness for Multinomial Models



n Histograms are composed from more multinomials than bins

q $X = w_1 B_1 + w_2 B_2 + w_3 B_3 + w_4 B_4 \dots$

n The mixture weights combining the multinomials are sparse

q I.e $\{w_i\}$ is sparse

q A *different* subset of weights w_i are high for different data

q Over a large collection of data vectors, all bases will eventually be used

Estimating Mixture Weights given Multinomials

n Basic estimation: Maximum likelihood

q $\text{Argmax}_W \log P(X ; B, W) = \text{Argmax}_W \sum_X X(f) \log(\sum_i w_i B_i(f))$

n Modified estimation: Maximum *a posteriori*

q $\text{Argmax}_W \sum_X X(f) (\sum_i w_i B_i(f)) + \log P(W)$

n Sparsity obtained by enforcing an *a priori* probability distribution $P(W)$ over the mixture weights that favors sparse mixture weights

n The algorithm for estimating weights must be modified to account for the priors

The *a priori* distribution

n A variety of *a priori* probability distributions all provide a bias towards “sparse” solutions

n The Dirichlet prior:

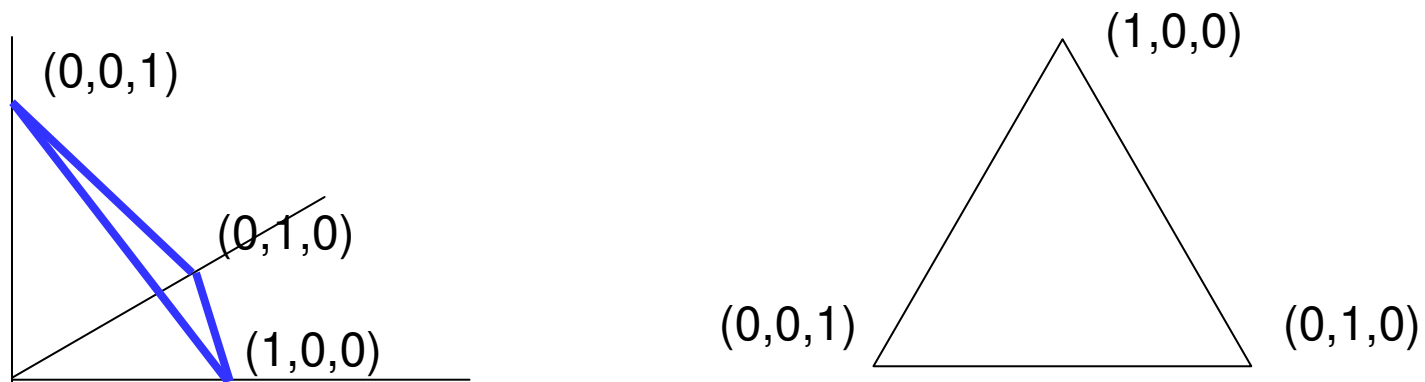
$$q \quad P(W) = Z^* \prod_i w_i^{a-1}$$

n The entropic prior:

$$q \quad P(W) = Z^* \exp(-aH(W))$$

$$n \quad H(W) = \text{entropy of } W = -\sum_i w_i \log(w_i)$$

A simplex view of the world

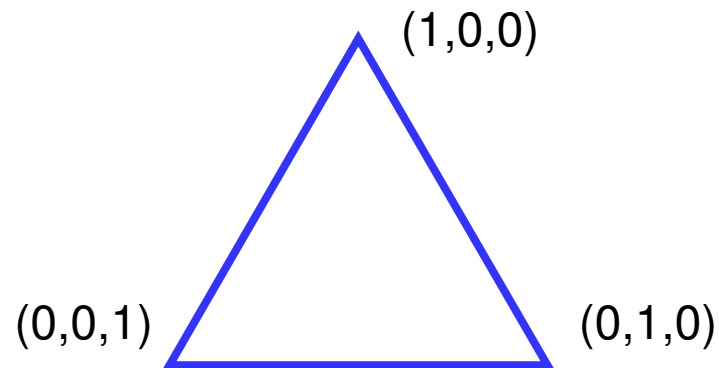


- n The mixture weights are a probability distribution
 - q $\sum_i w_i = 1.0$

- n They can be viewed as a vector
 - q $W = [w_0 \ w_1 \ w_2 \ w_3 \ w_4 \ \dots]$
 - q The vector components are positive and sum to 1.0

- n All probability vectors lie on a *simplex*
 - q A convex region of a linear subspace in which all vectors sum to 1.0

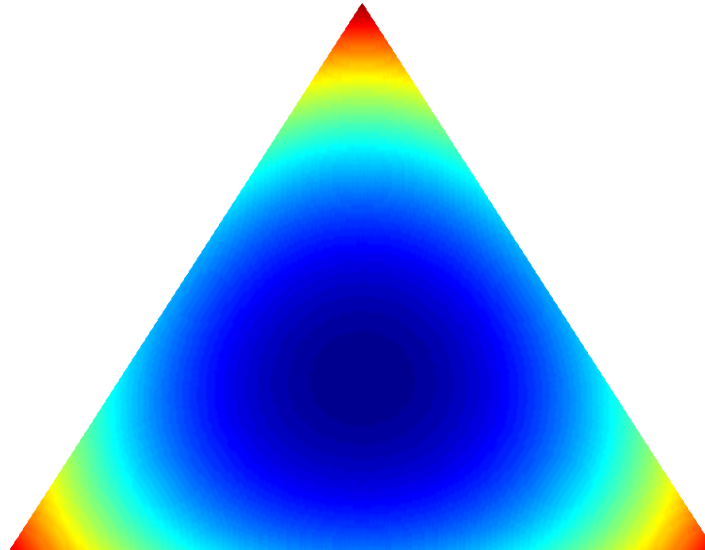
Probability Simplex



- n The sparsest probability vectors lie on the vertices of the simplex
- n The edges of the simplex are progressively less sparse
 - q Two-dimensional edges have 2 non-zero elements
 - q Three-dimensional edges have 3 non-zero elements
 - q Etc.

Sparse Priors: Dirichlet

2d Dirichlet Distribution Visualization Tool

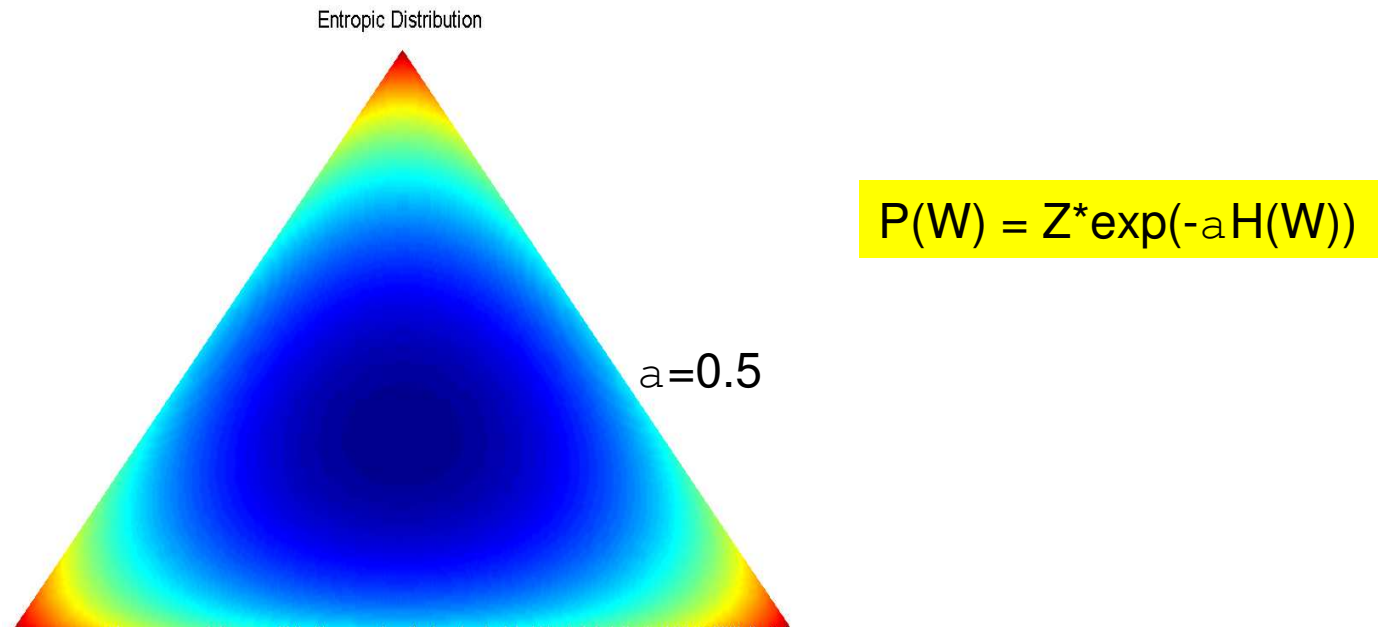


$\alpha=0.5$

$$P(W) = Z^* \prod_i w_i^{\alpha-1}$$

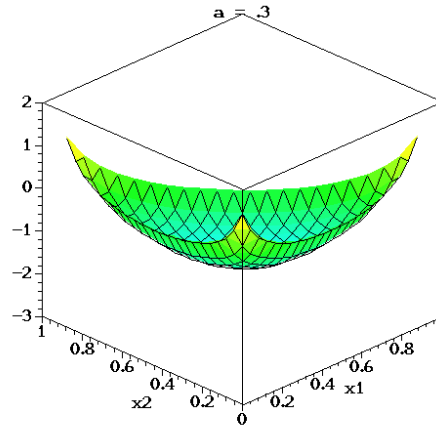
- n For $\alpha < 1$, sparse probability vectors are more likely than dense ones

Sparse Priors: The entropic prior



- n Vectors (probability distributions) with low entropy are more probable than those with high entropy
- q Low-entropy distributions are sparse!

The Entropic Prior



- n The entropic prior “controls” the desired level of sparsity in the mixture weights through a
- n Changing the sign of alpha can bias us towards either higher entropies or lower entropies

Optimization with the entropic prior

n The objective function

$$\text{Argmax}_W \sum_x X(f)(s_i w_i B_i(f)) - \alpha H(W)$$

n By estimating W such that the above equation is maximized, we can derive minimum entropy solutions

q Jointly optimize W for predicting the data while minimizing its entropy

The Expectation Maximization Algorithm

- n The parameters are actually learned using the *Expectation Maximization* (EM) algorithm
- n The EM algorithm actually optimizes the following objective function

$$Q = \sum_x P(Z | f) X(f) \log(P(Z) P(f|Z)) - aH(P(Z))$$

- n The second term here is derived from the entropic prior
- n Optimization of the above needs a solution to the following

$$\frac{\sum_f S(t, f) P_t(z | f)}{P_t(z)} + a(1 + \log P_t(z)) + 1 = 0$$

- n The solution requires a new function:
 - q The Lambert W function

Lambert's W Function

n Lambert's W function is the solution to:

$$W + \log(W) = X$$

q Where $W = F(X)$ is the Lambert function

n Alternately, the *inverse* function of

$$X = W \exp(W)$$

n In general, a multi-valued function

n If X is real, W is real for $X > -1/e$

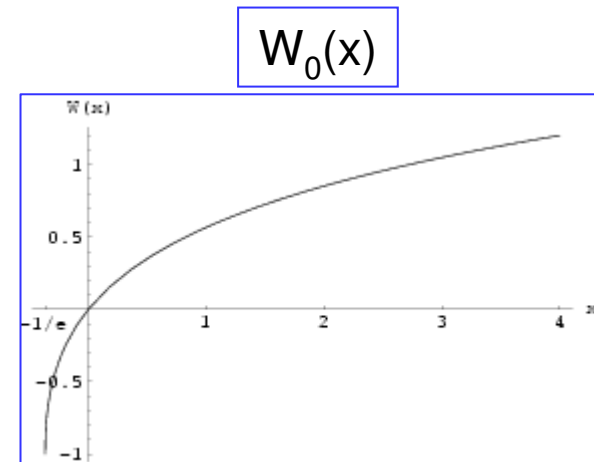
q Still multi-valued

n If we impose the restriction $W > -1$ and $W == \text{real}$ we get the zeroth branch of the W function

q Single valued

n For $W < -1$ and $W == \text{real}$ we get the -1th branch of the W function

q Single valued



Estimating $W_0(z)$

n An iterative solution

q Newton's Method

$$w_{j+1} = w_j - \frac{w_j e^{w_j} - z}{e^{w_j} + w_j e^{w_j}}$$

q Halley Iterations

$$w_{j+1} = w_j - \frac{w_j e^{w_j} - z}{e^{w_j}(w_j + 1) - \frac{(w_j + 2)(w_j e^{w_j} - z)}{2w_j + 2}}$$

q Code for Lambert's W function is available on wikipedia

Solutions with entropic prior

$$P_t(z) = \frac{-g/a}{W(-ge^{1+1/a} / a)}; \quad g = \sum_f S_t(f) P_t(z|f)$$

$$l = - \left(\frac{g}{P_t(z)} + a (1 + \log(P_t(z))) \right)$$

- n The update rules are the same as before, with one minor modification
- n To estimate the mixture weights, the above two equations must be iterated
 - q To convergence
 - q Or just for a few iterations
- n Alpha is the sparsity factor
- n $P_t(z)$ must be initialized randomly

Learning Rules for Overcomplete Basis Set

n Exactly the same as earlier, with the modification that $P_t(z)$ is now estimated to be sparse

q Initialize $P_t(z)$ for all t and $P(f|z)$

q Iterate

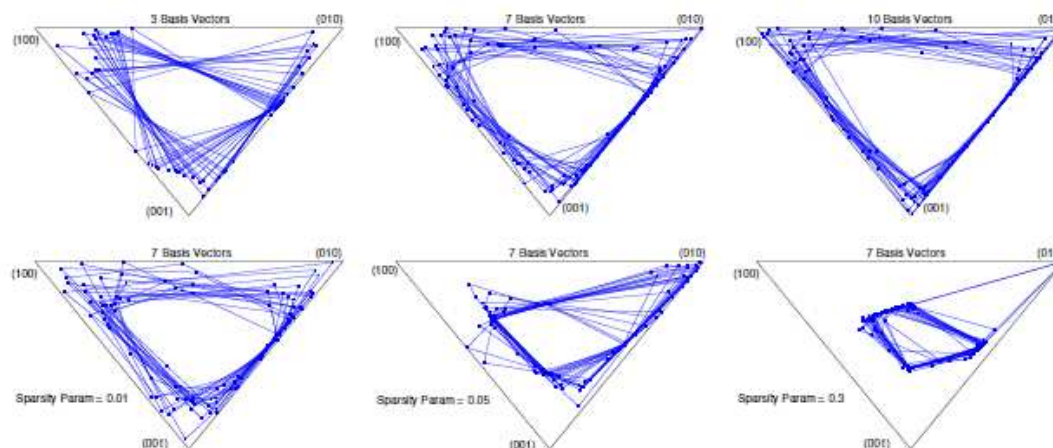
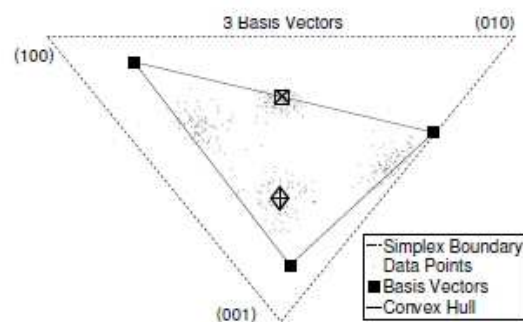
$$P_t(z|f) = \frac{P_t(z)P(f|z)}{\sum_{z'} P_t(z')P(f|z')}$$

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

$$P_t(z) = \frac{-g/a}{W(-ge^{1+1/a}/a)}; \quad g = \sum_f S_t(f)P_t(z|f)$$

$$l = -\left(\frac{g}{P_t(z)} + a(1 + \log(P_t(z))) \right)$$

A Simplex Example for Overcompleteness

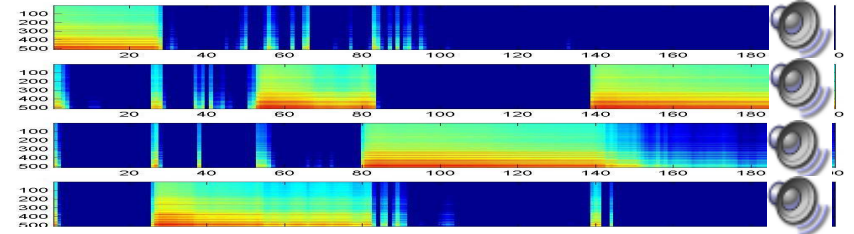
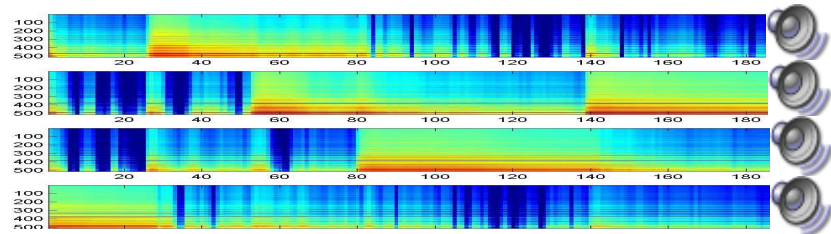
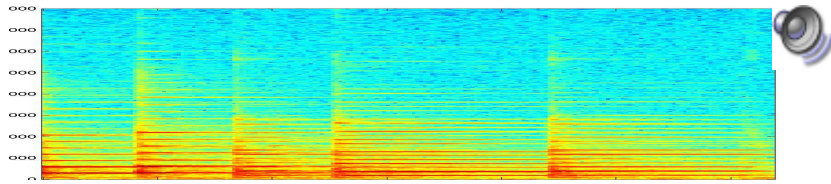
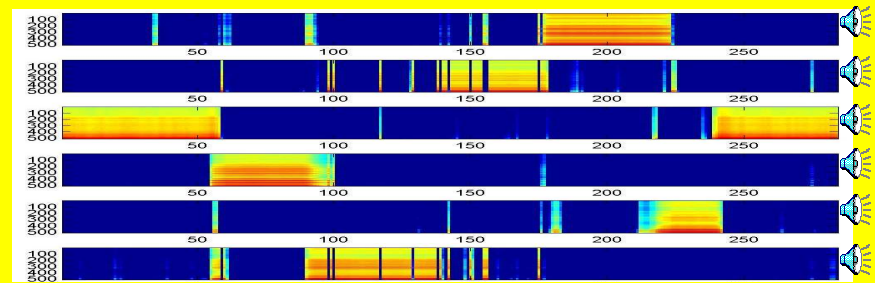
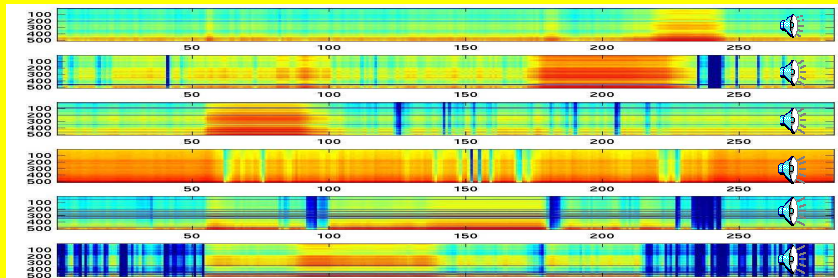
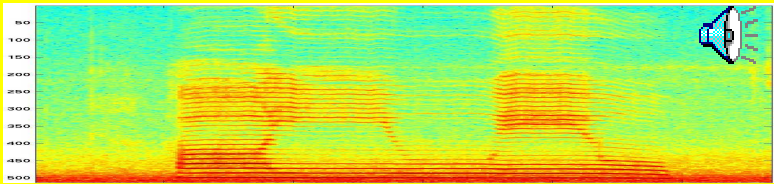


- n Synthetic data: Four clusters of data within the probability simplex
- n Regular learning with 3 bases learns an enclosing triangle
- n Overcomplete solutions without sparsity results in meaningless solutions
- n Sparse overcomplete model captures the distribution of the data

Sparsity can be employed *without* overcompleteness

- n Overcompleteness requires sparsity
- n Sparsity does *not* require overcompleteness
 - q Sparsity only imposes the constraint that the data are composed from a mixture of *as few multinomial components as possible*
 - q This makes no assumption about overcompleteness

Examples without overcompleteness



- n Left panel, Regular learning: most bases have significant energy in all frames
- n Right panel, Sparse learning: Fewer bases active within any frame
 - q Sparse decompositions result in more localized activation of bases
 - q Bases, too, are better defined in their structure

Face Data: The effect of sparsity

n As solutions get more sparse, bases become more informative

q In the limit, each basis is a complete face by itself.

q Mixture weights simply select face

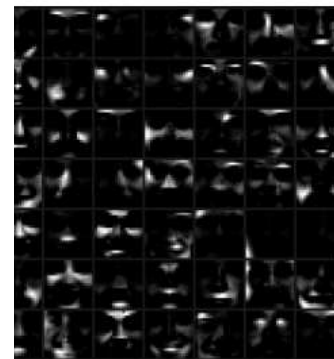
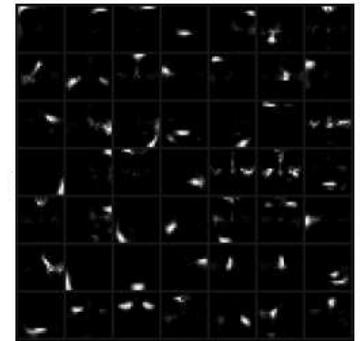
n Solution also allows for mixture weights to have *maximum* entropy

q *Maximally dense, i.e. minimally sparse*

q The bases become much more localized components

n The sparsity factor allows us to tune the bases we learn

High-entropy mixture weights

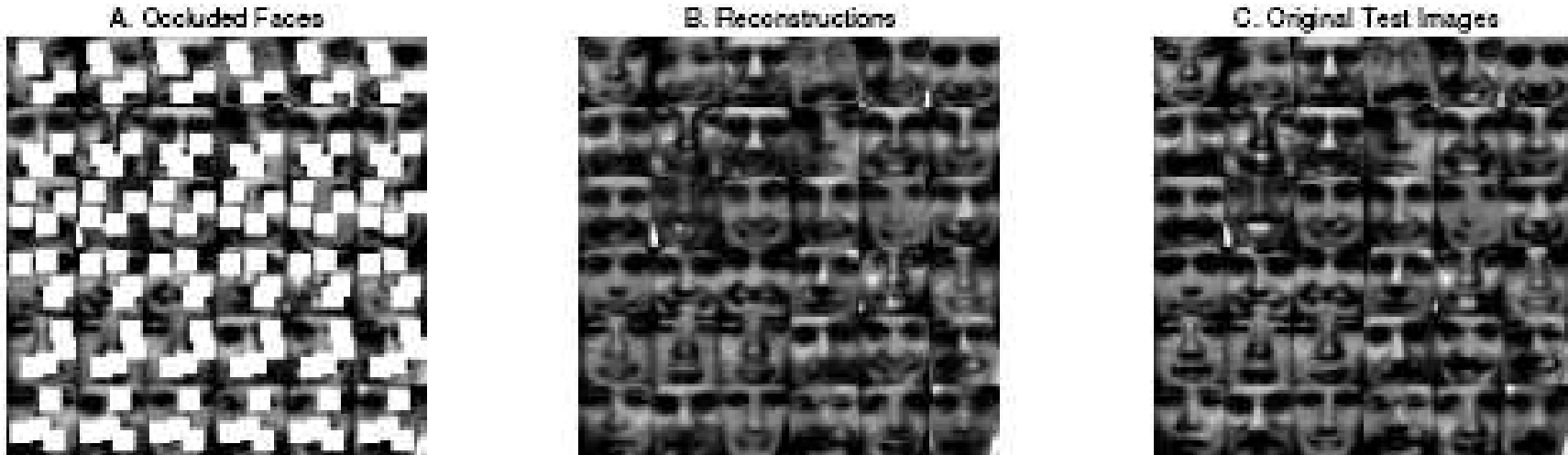


No sparsity

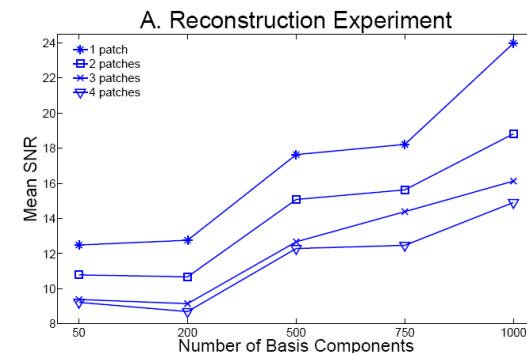


Sparse mixture weights

Benefit of overcompleteness



- n 19x19 pixel images (361 pixels)
- n Up to 1000 bases trained from 2000 faces
- n SNR of reconstruction from overcomplete basis set more than 10dB better than reconstruction from corresponding “compact” (regular) basis set

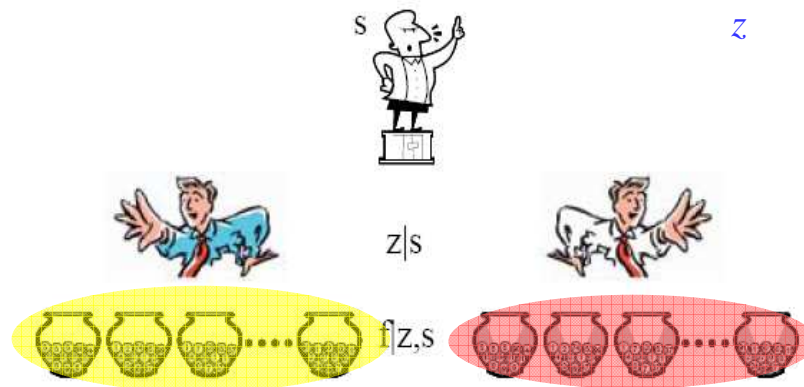


Signal Processing: How

- n Exactly as before
- n Learn an overcomplete set of bases
- n For each new data vector to be processed, compute the optimal mixture weights
 - q Constraining the mixture weights to be sparse now
- n Use the estimated mixture weights and the bases to perform additional processing

Signal Separation with Overcomplete Bases

- n Learn overcomplete bases for each source
- n For each frame of the mixed signal
 - q Estimate prior probability of source and mixture weights for each source
 - n Constraint: Use *sparse* learning for mixture weights
- n Estimate separated signals as $\hat{S}_{t,i}(f) = S_t(f) \sum_z P_t(z, s | f)$



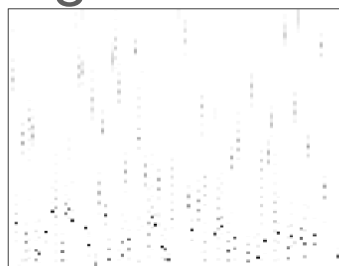
$$P_t(f) = P_t(s_1)P_t(f | s_1) + P_t(s_2)P_t(f | s_2)$$

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

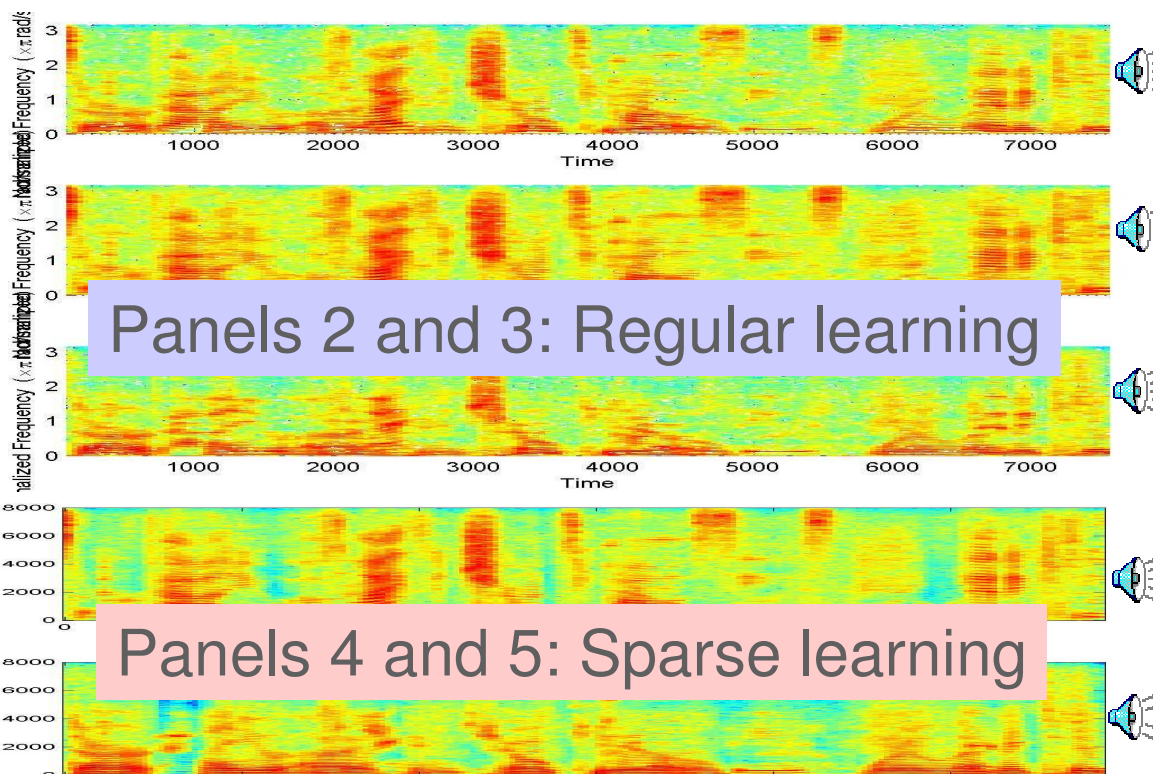
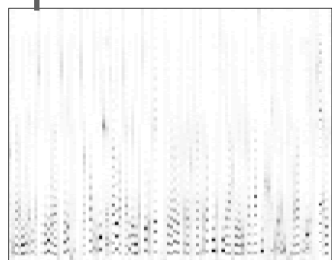
Sparse Overcomplete Bases: Separation

- n 3000 bases for each of the speakers
- q The speaker-to-speaker ratio typically doubles (in dB) w.r.t “compact” bases

Regular bases



Sparse bases



The Limits of Overcompleteness

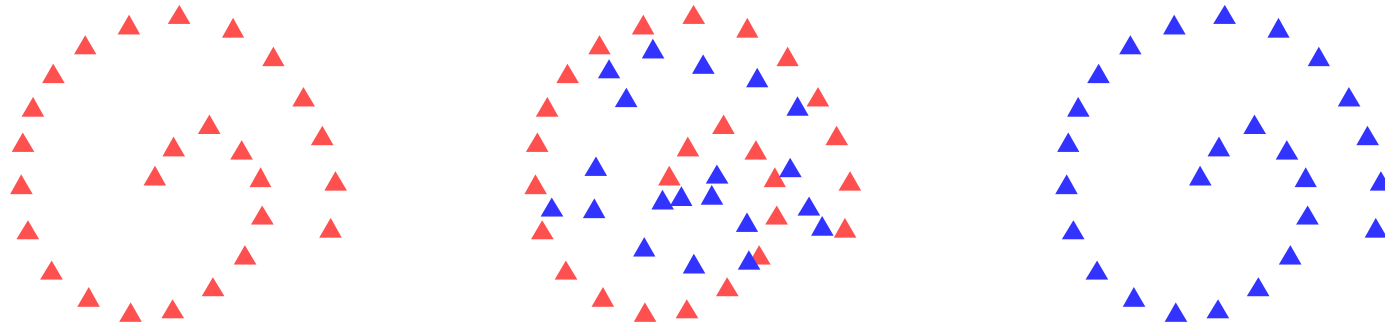
- n How many bases can we learn?
- n The limit is: as many bases as the number of vectors in the training data
 - q Or rather, the number of distinct histograms in the training data
 - n Since we treat each vector as a histogram
- n It is not possible to learn more than this number regardless of sparsity
 - q The arithmetic supports it, but the results will be meaningless

Working at the limits of overcompleteness:

The “Example-Based” Model

- n *Every training vector is a basis*
 - q Normalized to be a distribution
- n Let $S(t,f)$ be the t^{th} training vector
- n Let T be the total number of training vectors
- n The total number of bases is T
- n The k^{th} basis is given by
 - q $B(k,f) = S(k,f) / \sum_f S(k,f) = S(k,f) / |S(k,f)|_1$
- n Learning bases requires no additional learning steps besides simply collecting (and computing spectra from) training data

The example based model – an illustration

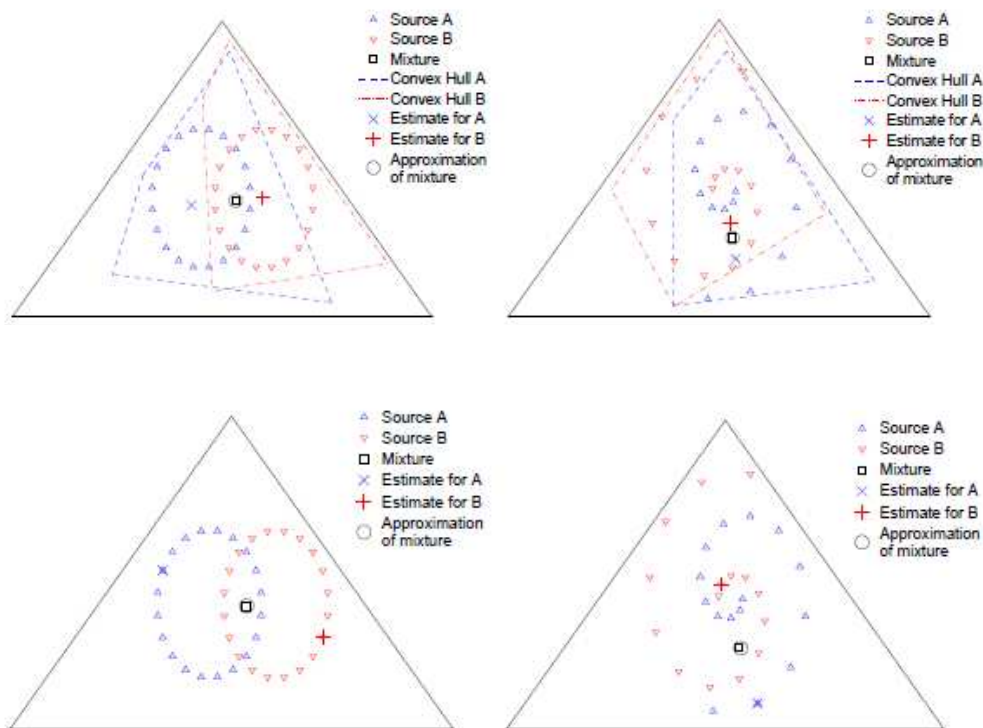


- n In the above example all training data lie on the curve shown (Left Panel)
 - q Each of them is a vector that sums to 1.0
- n The learning procedure for bases learns multinomial components that are linear combinations of the data (Middle Panel)
 - q These can lie anywhere within the area enclosed by the data
 - q The layout of the components hides the actual structure of the layout of the data
- n The example based representation captures the layout of the data perfectly (right panel)
 - q Since the data *are the bases*

Signal Processing with the Example Based Model

- n All previously defined operations can be performed using the example based model exactly as before
 - q For each data vector, estimate the optimal mixture weights to combine the bases
 - n Mixture weights MUST be estimated to be sparse
- n The example based representation is simply a special case of an overcomplete basis set

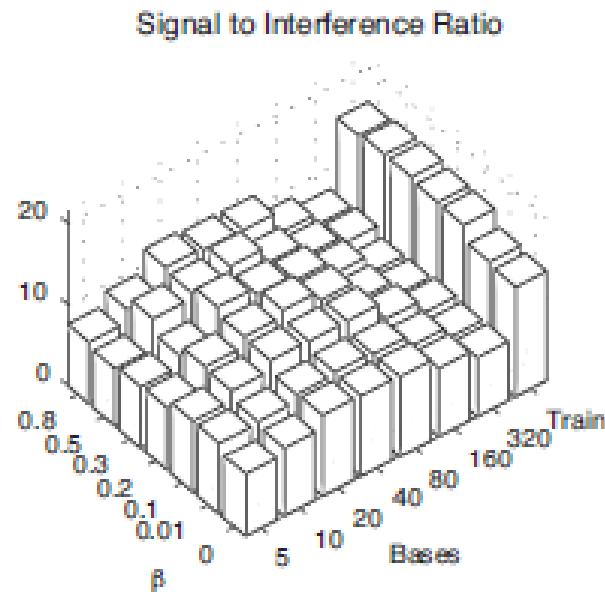
Illustrations of separation with example-based representation



n Top panel: Separation from learned bases

n Bottom panel: Separation with example-based representation

Speaker Separation Example



- n Speaker-to-interference ratio of separated speakers
- q State-of-the-art separation results

Example-based model: *All* the training data?

- n In principle, no need to use *all* training data as the model
 - q A well-selected subset will do
 - q E.g. – ignore spectral vectors from all pauses and non-speech regions of speech samples
 - q E.g. – eliminate spectral vectors that are nearly identical
- n The problem of *selecting* the optimal set of training examples remains open, however

Summary So Far

n PLCA:

- q The basic mixture-multinomial model for audio (and other data)

n Sparse Decomposition:

- q The notion of sparsity and how it can be imposed on learning

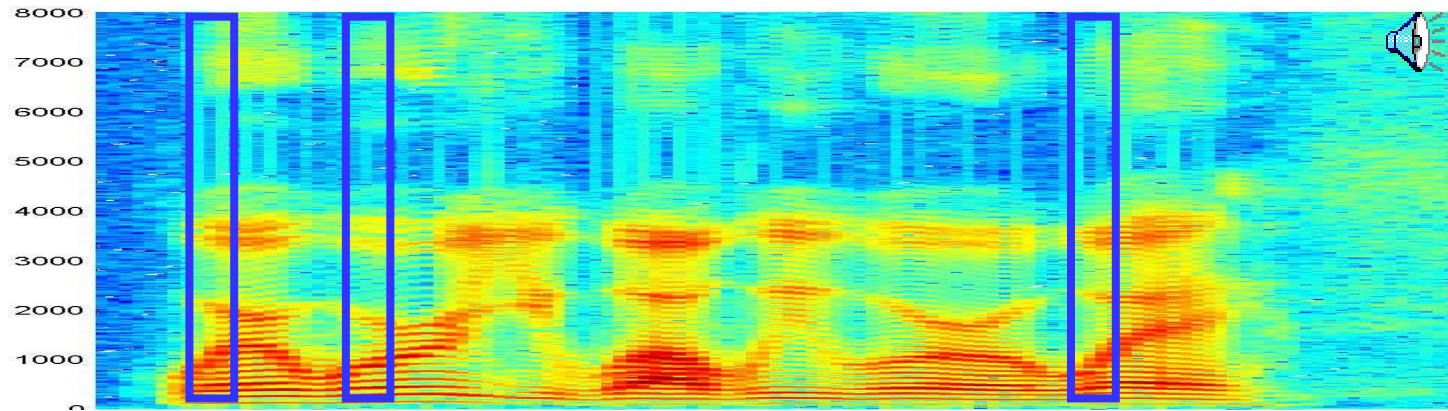
n Sparse Overcomplete Decomposition:

- q The notion of *overcomplete* basis set

n Example-based representations

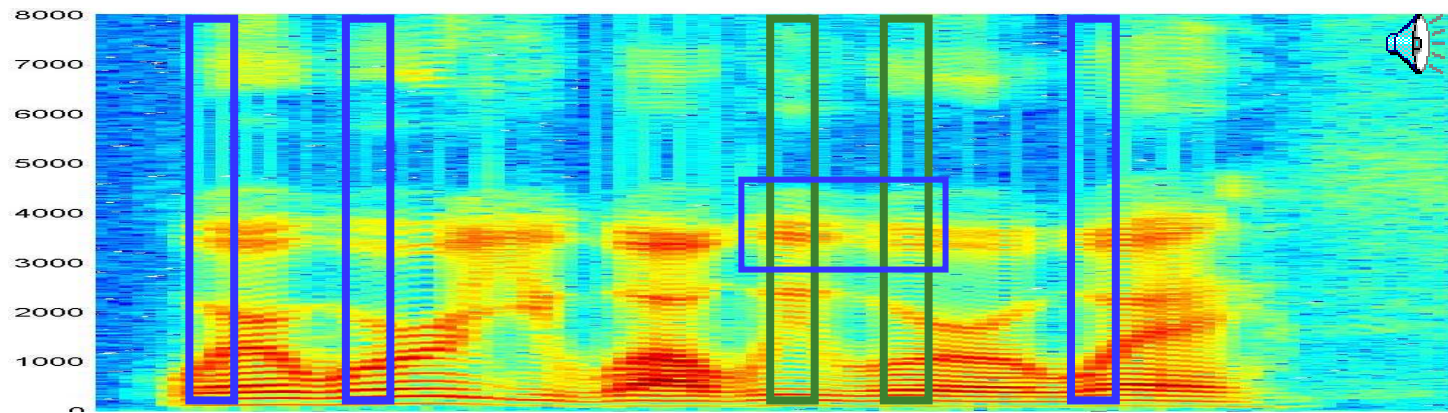
- q Using the training data itself as our representation

Next up: Shift/Transform Invariance



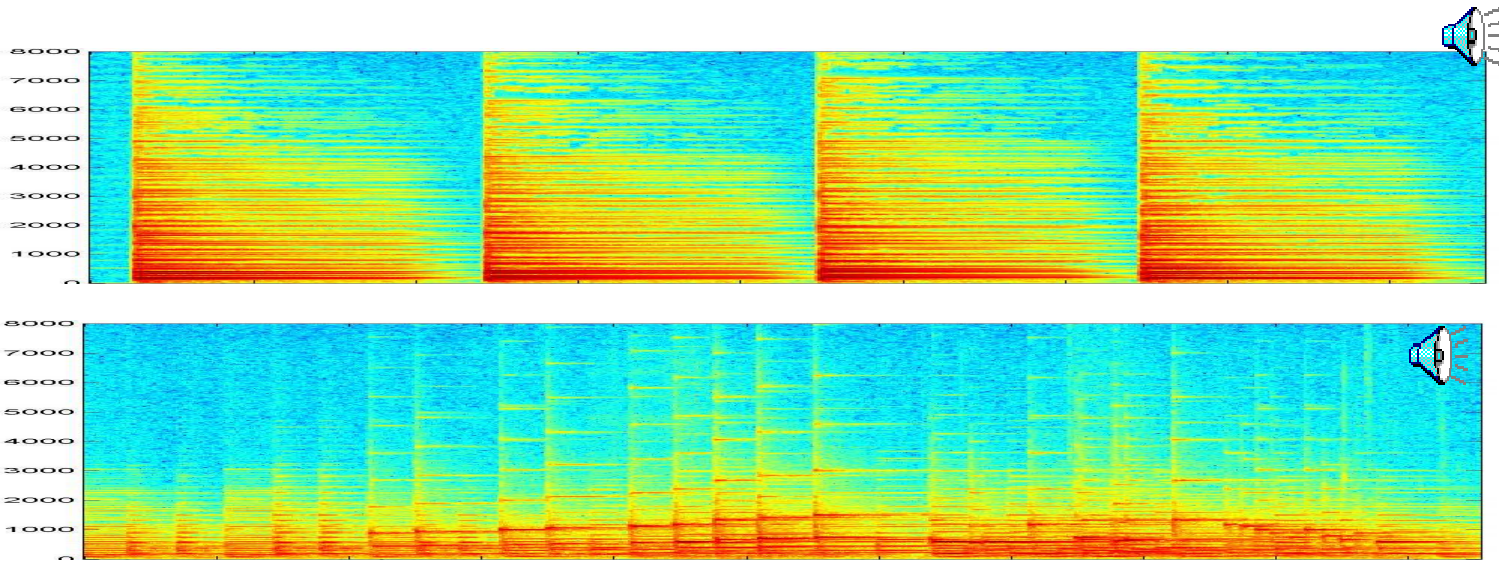
- n Sometimes the “typical” structures that compose a sound are wider than one spectral frame
- q E.g. in the above example we note multiple examples of a pattern that spans several frames

Next up: Shift/Transform Invariance



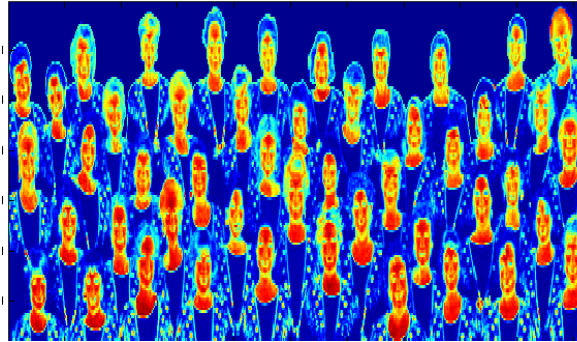
- n Sometimes the “typical” structures that compose a sound are wider than one spectral frame
 - q E.g. in the above example we note multiple examples of a pattern that spans several frames
- n Multiframe patterns may also be local in frequency
 - q E.g. the two green patches are similar only in the region enclosed by the blue box

Patches are more representative than frames



- n Four bars from a music example
- n The spectral patterns are actually patches
 - q Not all frequencies fall off in time at the same rate
- n The basic unit is a spectral patch, not a spectrum

Images: Patches often form the image

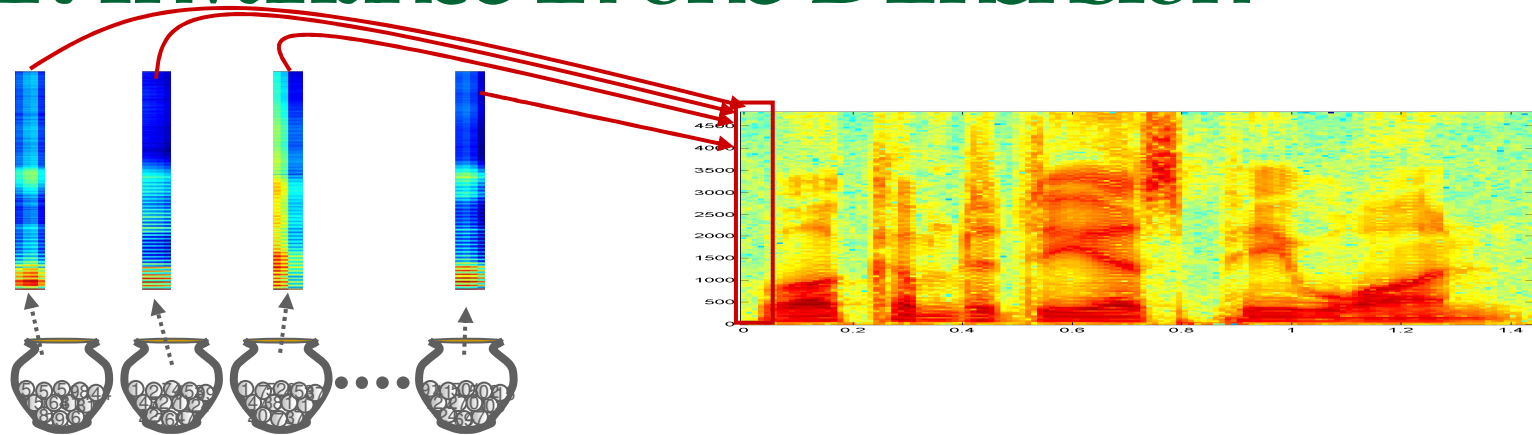


- n A typical image component may be viewed as a patch
 - q The alien invaders
 - q Face like patches
 - q A car like patch
- n overlaid on itself many times..

Shift-invariant modelling

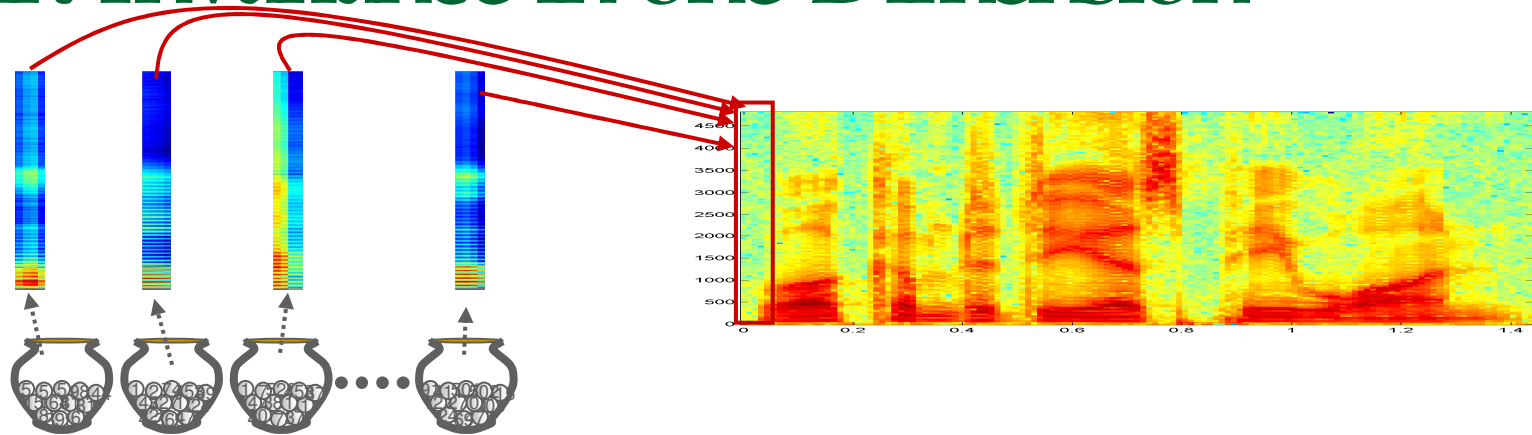
- n A shift-invariant model permits individual bases to be *patches*
- n Each patch composes the entire image.
- n The data is a sum of the compositions from individual patches

Shift Invariance in one Dimension



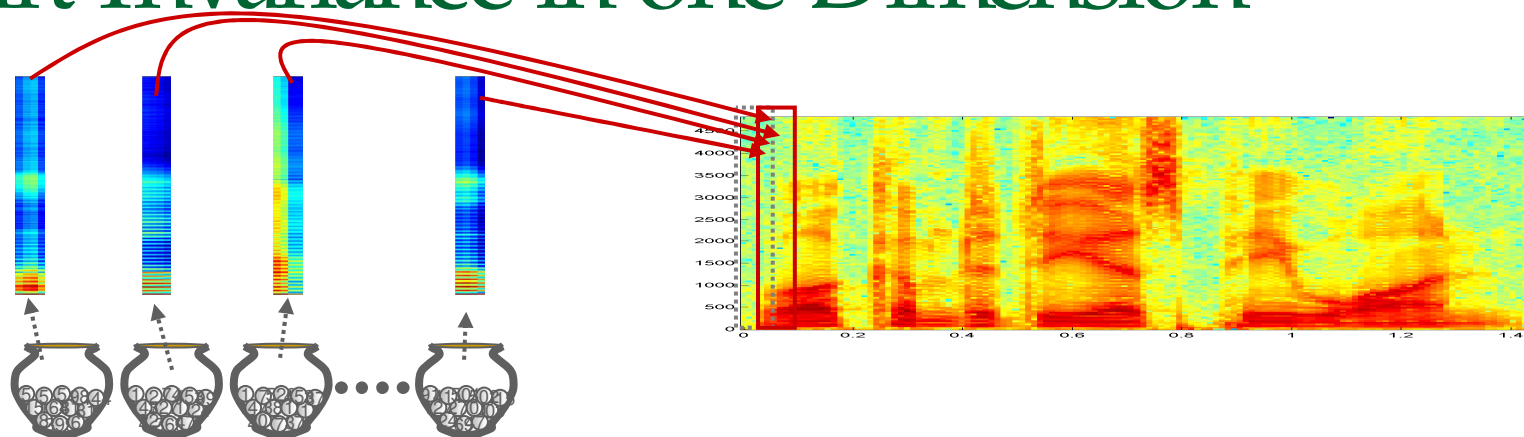
- n Our bases are now “patches”
 - q Typical *spectro-temporal* structures
- n The urns now represent patches
 - q Each draw results in a (t,f) pair, rather than only f
 - q *Also associated with each urn: A shift probability distribution $P(T|z)$*
- n The overall drawing process is slightly more complex
- n Repeat the following process:
 - q Select an urn Z with a probability $P(Z)$
 - q Draw a value T from $P(t|Z)$
 - q Draw (t,f) pair from the urn
 - q Add to the histogram at (t+T, f)

Shift Invariance in one Dimension



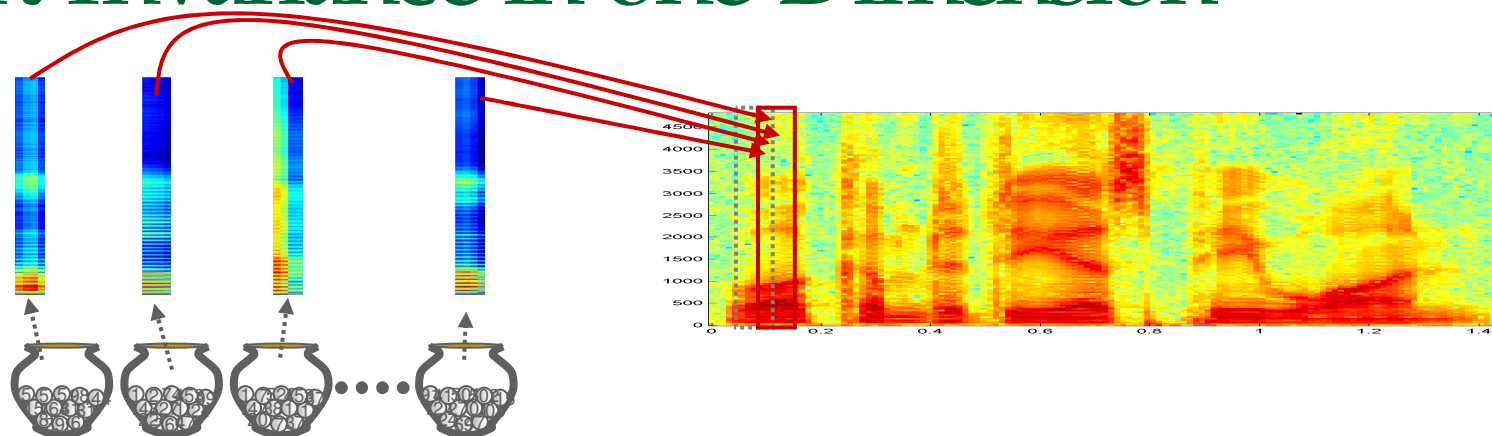
- n The process is *shift-invariant* because the probability of drawing a shift $P(T|Z)$ does not affect the probability of selecting urn Z
- n Every location in the spectrogram has contributions from every urn patch

Shift Invariance in one Dimension



- n The process is *shift-invariant* because the probability of drawing a shift $P(T|Z)$ does not affect the probability of selecting urn Z
- n Every location in the spectrogram has contributions from every urn patch

Shift Invariance in one Dimension



- n The process is *shift-invariant* because the probability of drawing a shift $P(T|Z)$ does not affect the probability of selecting urn Z
- n Every location in the spectrogram has contributions from every urn patch

Probability of drawing a particular (t,f) combination

$$P(t, f) = \sum_z P(z) \sum_t P(t | z) P(t - t, f | z)$$

- n The parameters of the model:
 - q $P(t, f | z)$ – the urns
 - q $P(T | z)$ – the *urn-specific* shift distribution
 - q $P(z)$ – probability of selecting an urn

- n The ways in which (t,f) can be drawn:
 - q Select any urn z
 - q Draw T from the urn-specific shift distribution
 - q Draw $(t-T, f)$ from the urn

- n The actual probability sums this over all shifts and urns

Learning the Model

- n The parameters of the model are learned analogously to the manner in which mixture multinomials are learned
- n Given observation of (t,f) , if we knew which urn it came from and the shift, we could compute all probabilities by counting!
 - q If shift is T and urn is Z
 - n $\text{Count}(Z) = \text{Count}(Z) + 1$
 - n For shift probability: $\text{Count}(T|Z) = \text{Count}(T|Z) + 1$
 - n For urn: $\text{Count}(t-T, f | Z) = \text{Count}(t-T, f|Z) + 1$
 - q Since the value drawn from the urn was $t-T, f$
 - q After all observations are counted:
 - n Normalize $\text{Count}(Z)$ to get $P(Z)$
 - n Normalize $\text{Count}(T|Z)$ to get $P(T|Z)$
 - n Normalize $\text{Count}(t,f|Z)$ to get $P(t,f|Z)$
- n Problem: When learning the urns and shift distributions from a histogram, the urn (Z) and shift (T) for any draw of (t,f) is not known
 - q These are unseen variables

Learning the Model

- n Urn Z and shift T are unknown
 - q So (t,f) contributes partial counts to every value of T and Z
 - q Contributions are proportional to the *a posteriori* probability of Z and T,Z

$$P(t, f, Z) = P(Z) \sum_T P(T | Z) P(t - T, f | Z) \quad P(T, t, f | Z) = P(T | Z) P(t - T, f | Z)$$

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

- n Each observation of (t,f)
 - q $P(z|t,f)$ to the count of the total number of draws from the urn
 - n $\text{Count}(Z) = \text{Count}(Z) + P(z | t,f)$
 - q $P(z|t,f)P(T | z,t,f)$ to the count of the shift T for the shift distribution
 - n $\text{Count}(T | Z) = \text{Count}(T | Z) + P(z|t,f)P(T | Z, t, f)$
 - q $P(z|t,f)P(T | z,t,f)$ to the count of (t-T, f) for the urn
 - n $\text{Count}(t-T, f | Z) = \text{Count}(t-T, f | Z) + P(z|t,f)P(T | z,t,f)$

Shift invariant model: Update Rules

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

$$P(t, f, Z) = P(Z) \sum_T P(T|Z) P(t-T, f|Z)$$

$$P(T, t, f | Z) = P(T|Z) P(t-T, f|Z)$$

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

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

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

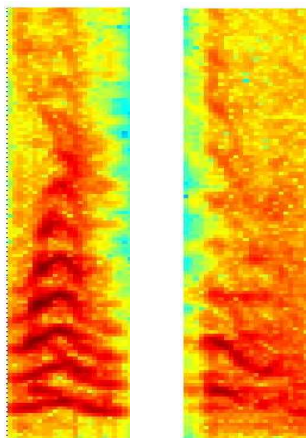
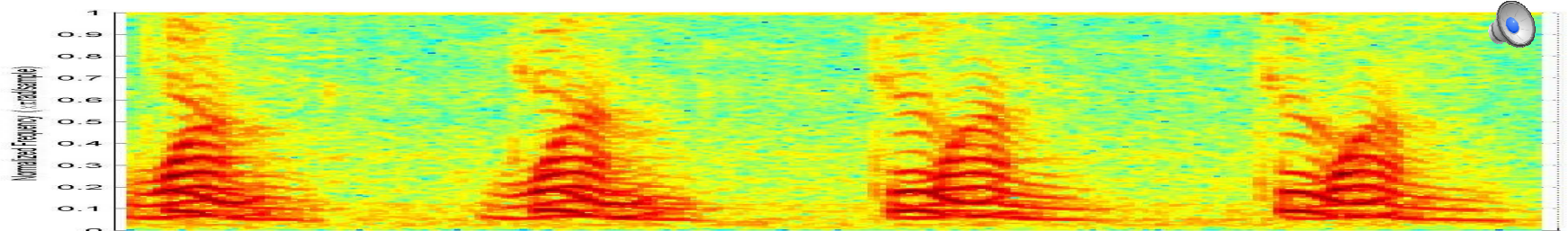
$$P(T | Z) = \frac{\sum_t \sum_f P(Z | t, f) P(T | Z, t, f) S(t, f)}{\sum_{T'} \sum_t \sum_f P(Z | t, f) P(T' | Z, t, f) S(t, f)}$$

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

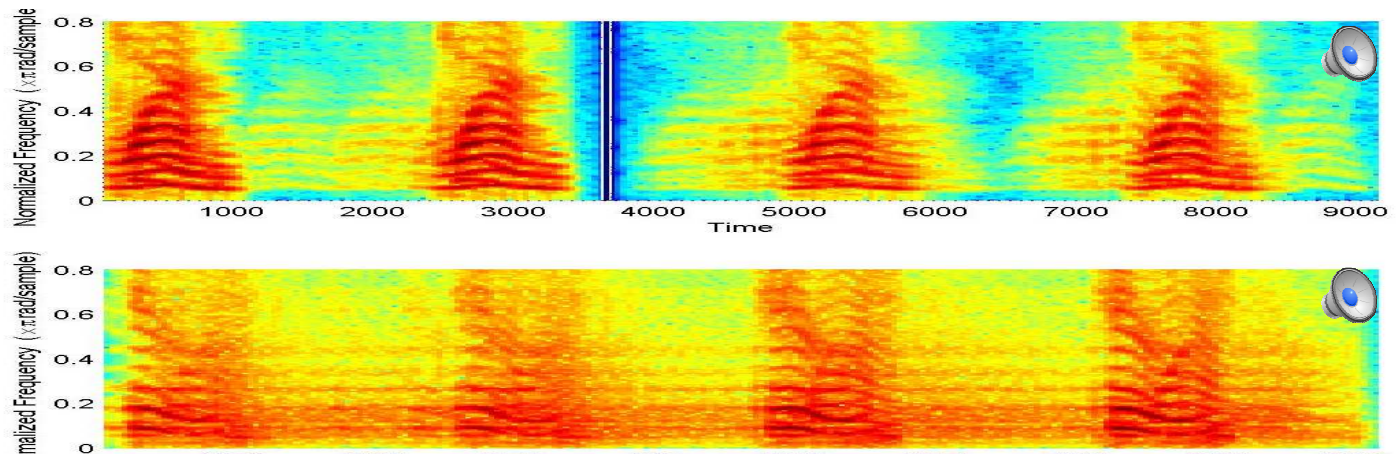
Shift-invariance in one time: example

- n An Example: Two distinct sounds occurring with different repetition rates within a signal
 - q Modelled as being composed from two time-frequency bases
 - q NOTE: Width of patches must be specified

INPUT SPECTROGRAM

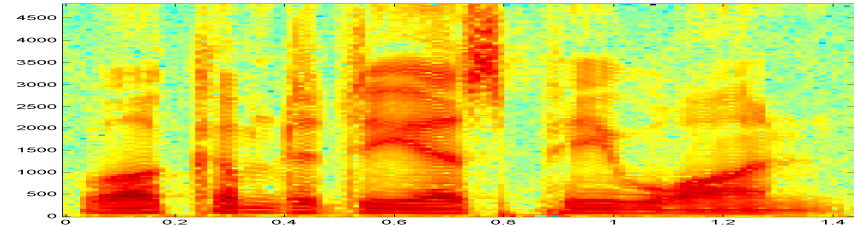
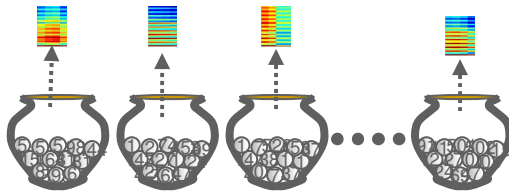


Discovered time-frequency
“patch” bases (urns)



Contribution of individual bases to the recording

Shift Invariance in Two Dimensions



- n We now have urn-specific shifts along both T and F
- n The Drawing Process
 - q Select an urn Z with a probability $P(Z)$
 - q Draw SHIFT values (T, F) from $P_s(T, F|Z)$
 - q Draw (t, f) pair from the urn
 - q Add to the histogram at $(t+T, f+F)$
- n This is a two-dimensional shift-invariant model
 - q We have shifts in both time and frequency
 - n Or, more generically, along both axes

Learning the Model

- n Learning is analogous to the 1-D case
- n Given observation of (t,f) , if we knew which urn it came from and the shift, we could compute all probabilities by counting!
 - q If shift is T,F and urn is Z
 - n $\text{Count}(Z) = \text{Count}(Z) + 1$
 - n For shift probability: $\text{ShiftCount}(T,F|Z) = \text{ShiftCount}(T,F|Z) + 1$
 - n For urn: $\text{Count}(t-T,f-F | Z) = \text{Count}(t-T,f-F|Z) + 1$
 - q Since the value drawn from the urn was $t-T,f$
 - q After all observations are counted:
 - n Normalize $\text{Count}(Z)$ to get $P(Z)$
 - n Normalize $\text{ShiftCount}(T,F|Z)$ to get $P_s(T,F|Z)$
 - n Normalize $\text{Count}(t,f|Z)$ to get $P(t,f|Z)$
- n Problem: Shift and Urn are unknown

Learning the Model

- n Urn Z and shift T, F are unknown
 - q So (t, f) contributes partial counts to every value of T, F and Z
 - q Contributions are proportional to the *a posteriori* probability of Z and $T, F | Z$

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

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

- n Each observation of (t, f)
 - q $P(z | t, f)$ to the count of the total number of draws from the urn
 - n $\text{Count}(Z) = \text{Count}(Z) + P(z | t, f)$
 - q $P(z | t, f) P(T, F | z, t, f)$ to the count of the shift T, F for the shift distribution
 - n $\text{ShiftCount}(T, F | Z) = \text{ShiftCount}(T, F | Z) + P(z | t, f) P(T | Z, t, f)$
 - q $P(T | z, t, f)$ to the count of $(t - T, f - F)$ for the urn
 - n $\text{Count}(t - T, f - F | Z) = \text{Count}(t - T, f - F | Z) + P(z | t, f) P(t - T, f - F | z, t, f)$

Shift invariant model: Update Rules

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

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

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

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

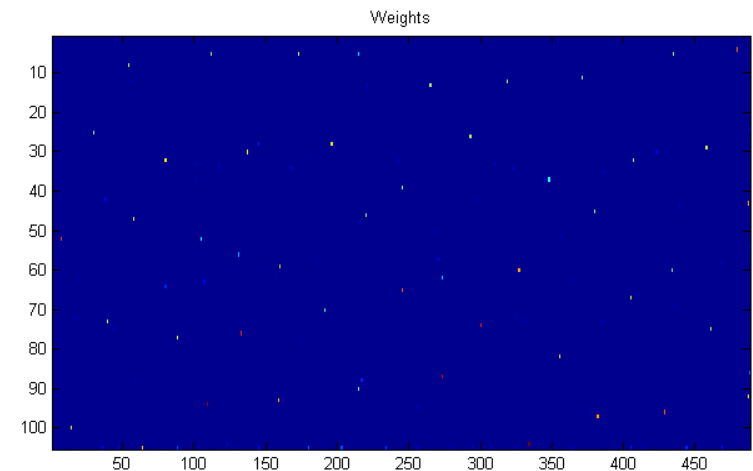
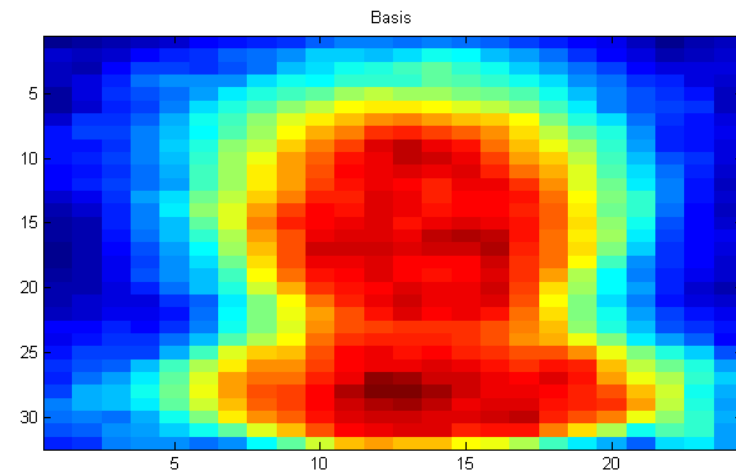
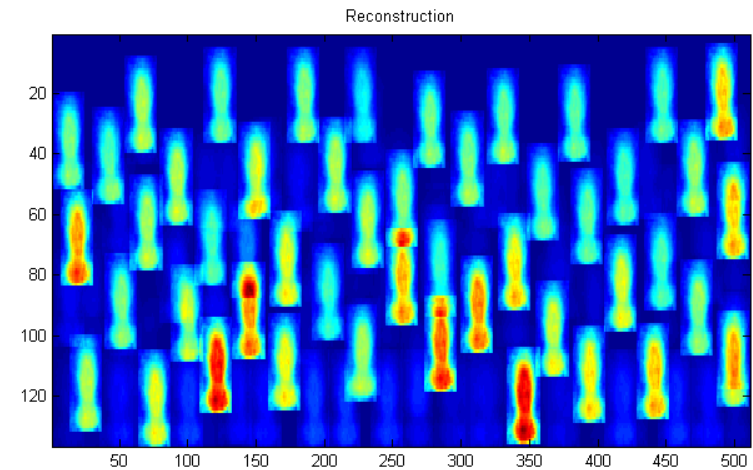
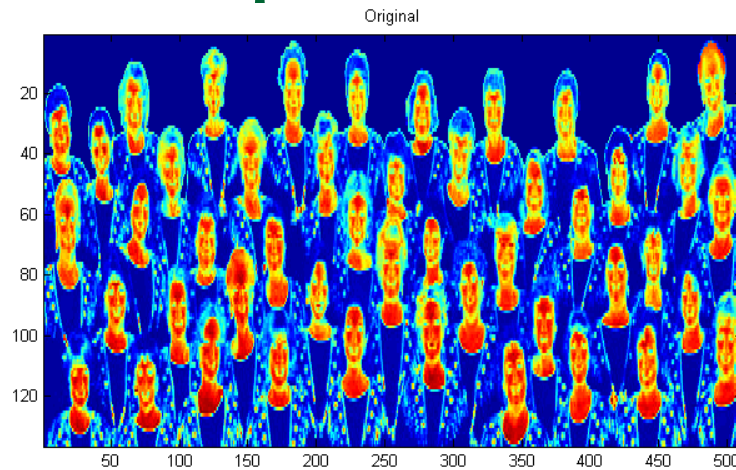
$$P(Z) = \frac{\sum_t \sum_f P(Z | t, f) S(t, f)}{\sum_{Z'} \sum_t \sum_f P(Z' | t, f) S(t, f)} \quad P(T, F | Z) = \frac{\sum_t \sum_f P(Z | t, f) P(T, F | Z, t, f) S(t, f)}{\sum_{T'} \sum_{F'} \sum_t \sum_f P(Z | t, f) P(T', F' | Z, t, f) S(t, f)}$$

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

2D Shift Invariance: The problem of indeterminacy

- n $P(t,f|Z)$ and $P_s(T,F|Z)$ are analogous
 - q Difficult to specify which will be the “urn” and which the “shift”
- n Additional constraints required to ensure that one of them is clearly the shift and the other the urn
- n Typical solution: Enforce sparsity on $P_s(T,F|Z)$
 - q The patch represented by the urn occurs only in a few locations in the data

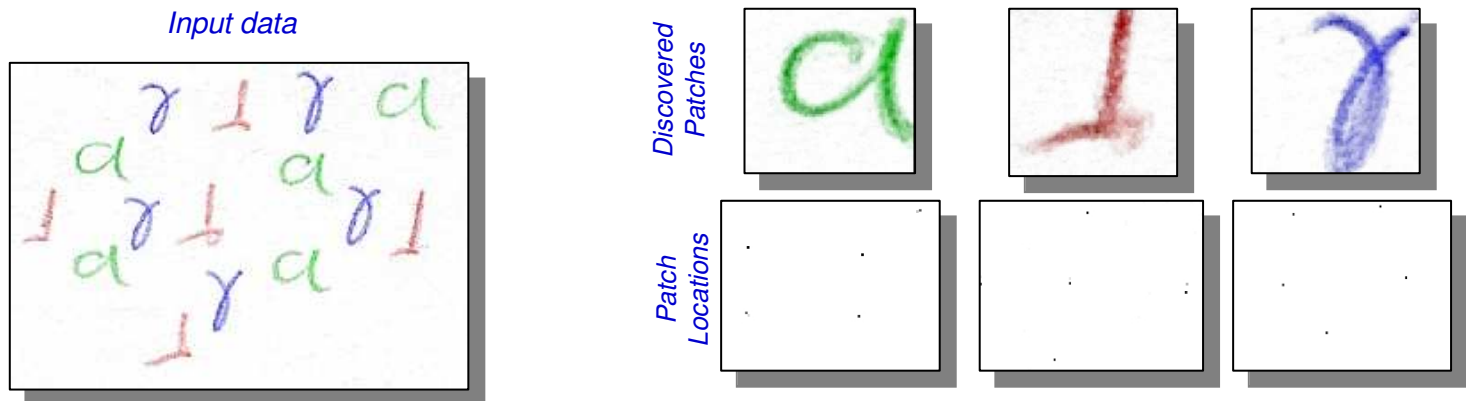
Example: 2-D shift invariance



- n Only one “patch” used to model the image (i.e. a single urn)
- q The learnt urn is an “average” face, the learned shifts show the locations of faces

Example: 2-D shift invariance

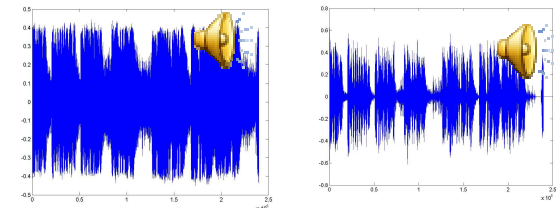
- n The original figure has multiple handwritten renderings of three characters
 - q In different colours
- n The algorithm learns the three characters and identifies their locations in the figure



Shift-Invariant Decomposition – Uses

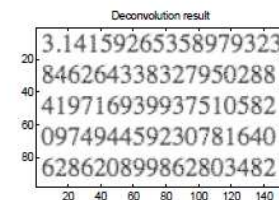
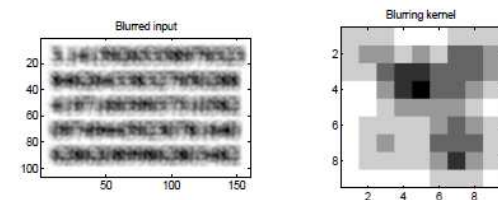
n Signal separation

- q The arithmetic is the same as before
- q Learn shift-invariant bases for each source
- q Use these to separate signals



n Dereverberation

- q The spectrogram of the reverberant signal is simply the sum several shifted copies of the spectrogram of the original signal
- n 1-D shift invariance

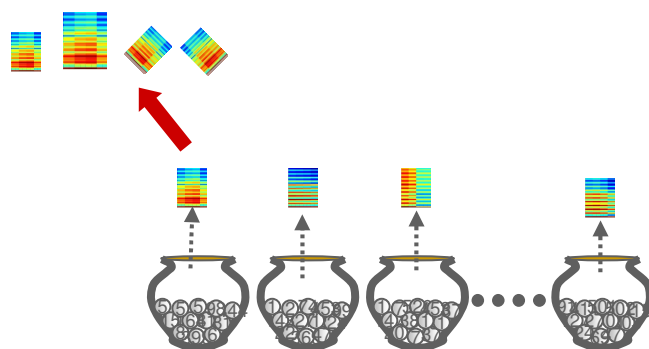


n Image Deblurring

- q The blurred image is the sum of several shifted copies of the clean image
- n 2-D shift invariance

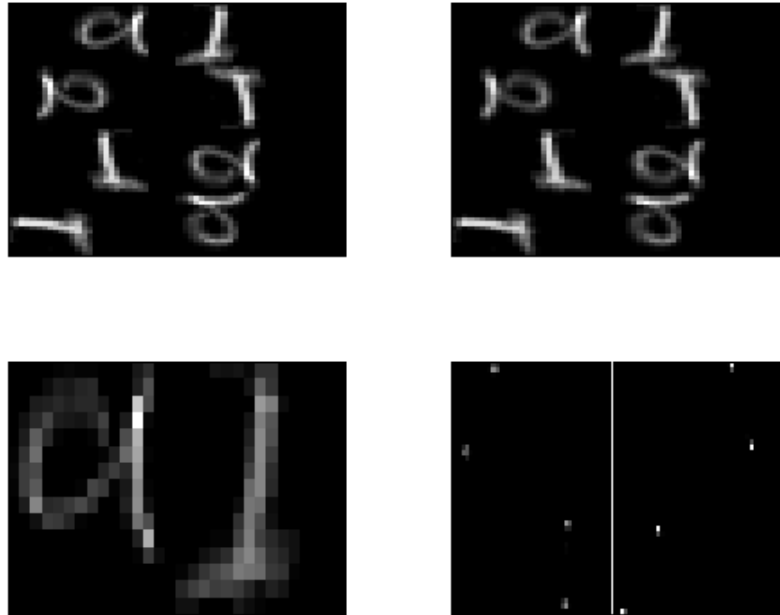


Beyond shift-invariance: transform invariance



- n The draws from the urns may not only be shifted, but also transformed
- n The arithmetic remains very similar to the shift-invariant model
 - q We must now impose one of an enumerated set of transforms to (t, f) , after shifting them by (T, F)
 - q In the estimation, the precise transform applied is an unseen variable

Example: Transform Invariance



- n Top left: Original figure
- n Bottom left – the two bases discovered
- n Bottom right –
 - q Left panel, positions of “a”
 - q Right panel, positions of “l”
- n Top right: estimated distribution underlying original figure

Transform Invariance: Uses and Limitations

- n Not very useful to analyze audio
- n May be used to analyze images and video
- n Main restriction: Computational complexity
 - q Requires unreasonable amounts of memory and CPU
 - q Efficient implementation an open issue

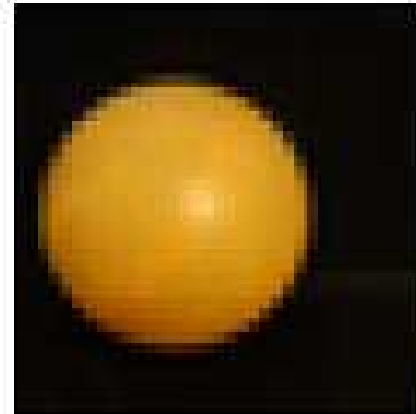
Example: Higher dimensional data

n Video example

Description of Input



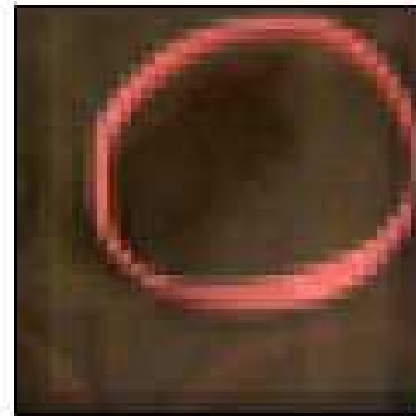
Kernel 1



Kernel 2



Kernel 3



Summary

- n Shift invariance

- q Multinomial bases can be “patches”

- n Representing time-frequency events in audio or other larger patterns in images

- n Transform invariance

- q The patches may further be transformed to compose an image

- n Not useful for audio