4 Model-Based Scene Analysis

DANIEL P. W. ELLIS

4.1 INTRODUCTION

When multiple sound sources are mixed together into a single channel (or a small number of channels) it is in general impossible to recover the exact waveforms that were mixed – indeed, without some kind of constraints on the form of the component signals it is impossible to separate them at all. These constraints could take several forms: for instance, given a particular family of processing algorithms (such as linear filtering, or selection of individual time-frequency cells in a spectrogram), constraints could be defined in terms of the relationships between the set of resulting output signals, such as statistical independence [3, 41], or clustering of a variety of properties that indicate distinct sources [45, 1]. These approaches are concerned with the relationships between the properties of the complete set of output signals, rather than the specific properties of any individual output; in general, the individual sources could take any form.

But another way to express the constraints is specify the form that the individual sources can take, regardless of the rest of the signal. These restrictions may be viewed as 'prior models' for the sources, and source separation then becomes the problem of finding a set of signals that combine together to give the observed mixture signal at the same time as conforming in some optimal sense to the prior models. This is the approach to be examined in this chapter.

4.2 SOURCE SEPARATION AS INFERENCE

It is helpful to start with a specific description. We will consider this search for well-formed signals from a probabilistic point of view: Assume an observed mixture signal $x(t)$ is composed as the sum of a set of source signals $\{s_i(t)\}$ (where the

Computational Auditory Scene Analysis. Edited by DeLiang Wang and Guy J. Brown

braces denote a set, and each s_i is an individual source) via

$$x(t) = \sum_i s_i(t) + \nu(t) \tag{4.1}$$

where $\nu(t)$ is a noise signal (comprising sensor and/or background noise). If for example we assume ν is Gaussian and white with variance σ^2, then the probability density function (pdf) for x becomes a normal distribution whose mean is the sum of the sources,

$$p(x(t)|\{s_i\}) = \mathcal{N}(x(t); \sum_i s_i(t), \sigma^2) \tag{4.2}$$

i.e. $x(t)$ is Gaussian-distributed with mean $\sum_i s_i(t)$ and variance σ^2. (In our notation, $x(t)$ refers to the scalar random value at a particular time, whereas omitting the time dependency to leave x refers to the complete function for all time. Thus the observation at a particular time, $x(t)$, may in general depend on the source signals at other times, which are included in s_i; for this particular example, however, $x(t)$ depends only on the values of the sources at that same instant, $\{s_i(t)\}$.) Different noise characteristics would give us different pdfs, but the principle is the same. Now, we can apply Bayes rule to get:

$$p(\{s_i\}|x) = \frac{p(x|\{s_i\})p(\{s_i\})}{p(x)} \tag{4.3}$$

which introduces prior probability terms for both the observation $p(x)$ and for the set of component sources $p(\{s_i\})$. If we assume that the sources are more or less independent of one another, we can factor $p(\{s_i\}) = \prod_i p(s_i)$ i.e. the product of independent prior distributions for each source.

Eqn. 4.3 is a precise statement of our source separation problem: we have an observed signal x and we want to infer what we can about the individual sources s_i it contains. If the posterior distribution over the set $\{s_i\}$ is broad and equivocal, we do not have enough information to recover the sources from the mixture, but if there are compact regions where particular values for $\{s_i\}$ have a large posterior probability, we can make useful inferences that the original source signals are most likely as predicted by those regions. If pressed to recover estimates of actual source signals (rather than simply expressing our uncertainty about them), we can search for the most likely sources given the observations,

$$\{\hat{s_i}\} = \underset{\{s_i\}}{\operatorname{argmax}}\; p(\{s_i\}|x) = \underset{\{s_i\}}{\operatorname{argmax}}\; p(x|\{s_i\}) \prod_i p(s_i) \tag{4.4}$$

where $p(x)$ has disappeared since it does not influence the search for the maximum.

In theory, then, source separation could be performed by searching over all possible combinations of source signals to find the most likely set; in practice, this is an exponential search space over continuous signals, so direct search is impractical. Eqn. 4.4 can, however, be seen as a very general expression of the source separation problem, decomposed into two pieces: $p(x|\{s_i\})$ i.e. to what extent the proposed components $\{s_i\}$ are consistent with the observations, and $p(s_i)$, the *a priori* likelihood of each candidate source signal. $p(x|\{s_i\})$ is relatively simple to compute given a complete set of source components (eqn. 4.2), so the power in solving this problem, if it can be solved, needs to come from $p(s_i)$, the likelihood of a particular signal s_i under our *source model*. To emphasize the role of this model by giving it a symbol M, we can write this probability as $p(s_i|M)$.

$p(s_i|M)$ embodies the constraints to the effect that s_i cannot be just any signal (if it could, the source separation problem would generally be insoluble), but instead is limited to take on some particular forms. Hence, even though mixing the sources into x has irretrievably lost dimensionality (e.g. from the $N \times T$ samples of N sources at T discrete time steps, down to just T samples of x), there may still be a unique, or tightly-bounded, set of sources $\{s_i\}$ consistent both with the observed x and the prior model constraints $p(s_i|M)$.

As an example, consider the situation where observations consist of multidimensional vectors – think, perhaps, of the short-time Fourier transform (STFT) magnitude columns of a spectrogram – and each source is constrained to consist of a sequence of frames drawn from a dictionary; in Roweis [36] these dictionaries are represented as vector-quantizer codebooks trained to capture the range of short-time spectral characteristic of a particular voice. Then separating a mixture of two sources is a finite search at each time step over all entries from the two codebooks to find the pair of vectors that combine to match the observation most closely, subject to learned and/or assumed uncertainty and noise.

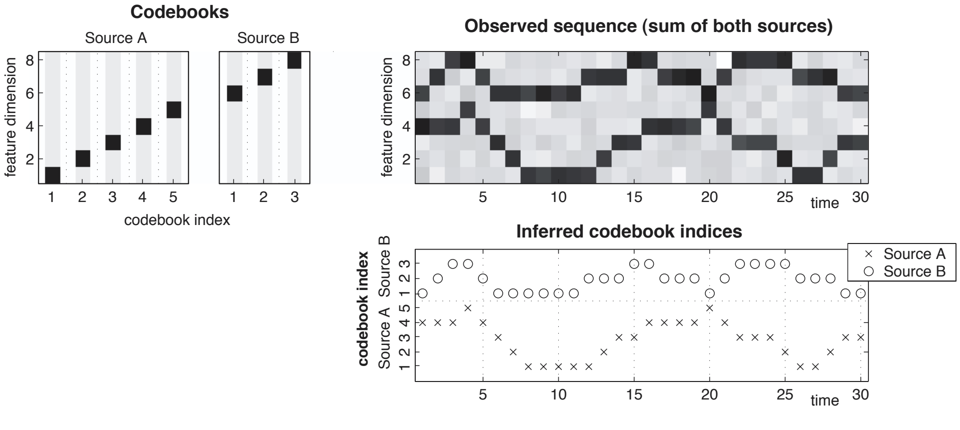


Fig. 4.1 Example of a mixture composed of two sources defined by codebooks as shown on the left. The codebook states are distinct between the two sources, and the individual source states are readily inferred from the observed mixture.

Figure 4.1 illustrates this case. In an 8-dimensional feature vector, the signal emitted by source A can be described with just 5 prototype slices (that may occur in any order), and source B with three more. In the illustration, each prototype slice consists of a vector that is flat except for one dimension that is larger than the rest. Moreover, the dominant dimensions are taken from different subsets for the two sources. Thus, in the noisy observations of a mixture (sum) of signals from both sources, it is trivial to identify which state from each source provides the best fit to each time frame: the choices are unambiguous, both between and within sources. In terms of eqn. 4.4, the likelihood of a ‘fit’ of a possible isolated source signal s_i under the source model is simply a product of the likelihoods of each feature vector (since the model includes no sequential constraints). The behavior of each source s_i is defined by the state model denoted as M_i, and which we can introduce explicitly into the equations by making it a conditioning term i.e. $p(s_i) \equiv p(s_i|M_i)$. For the particular case of these memoryless codebooks, we have

$$\begin{aligned} p(s_i|M_i) &= \prod_t p\left(s_i(t)|M_i\right) \\ &= \prod_t \sum_{q_i(t)} p\left(s_i(t)|q_i(t)\right) p\left(q_i(t)|M_i\right) \end{aligned} \tag{4.5}$$

where $q_i(t)$ is the (discrete-valued) codebook state index for model M_i at time t; specifying $q_i(t)$ captures everything we need to know about the model M_i to calculate $p(s_i(t))$, as indicated by the disappearance of M_i from the conditioning term for $s_i(t)$ once $q_i(t)$ is present. $p(q_i(t)|M_i)$ is a prior weight that can bias the model towards particular codewords, but we can assume it is essentially uniform for allowed codewords. We could express the codebook relationship as:

$$p\left(s_i(t)|q_i(t)\right) = \mathcal{N}(s_i(t); C_i\left(q_i(t)\right), \Sigma_i) \tag{4.6}$$

where C_i is the codebook lookup table of prototype slices, and covariance Σ_i allows each codeword to cover a small volume of signal space rather than a discrete point; Σ_i could also be taken from a lookup table indexed by $q_i(t)$, or could be shared across the whole codebook as suggested in eqn. 4.6. As the elements of Σ_i reduce to zero, s_i becomes deterministically related to the codebook index, $p\left(s_i(t)|q_i(t)\right) = \delta(s_i(t) - C_i\left(q_i(t)\right))$ (the Dirac delta function), or simply $s_i(t) = C_i\left(q_i(t)\right)$.

If we assume the codewords are arranged to cover essentially non-overlapping regions of signal space, then for any possible value of $s_i(t)$, there will be just one codebook index that dominates the summation in eqn. 4.5, so the summation can be well approximated by the single term for that index:

$$\begin{aligned} p(s_i|M_i) &= \prod_t \sum_{q_i(t)} p\left(s_i(t)|q_i(t)\right) p\left(q_i(t)|M_i\right) \\ &\approx \prod_t \max_{q_i(t)} p\left(s_i(t)|q_i(t)\right) p\left(q_i(t)|M_i\right) \\ &= \prod_t p\left(s_i(t)|q_i^*(t)\right) p\left(q_i^*(t)|M_i\right) \end{aligned} \tag{4.7}$$

where the best state index at each time

$$q_i^*(t) = \underset{q_i(t)}{\text{argmax}}\ p\left(s_i(t)|q_i(t)\right) p\left(q_i(t)|M_i\right) \tag{4.8}$$

or, assuming a spherical covariance for the codewords and uniform priors,

$$q_i^*(t) = \underset{q_i(t)}{\text{argmin}}\ \|s_i(t) - C_i\left(q_i(t)\right)\| \tag{4.9}$$

Thus, given the individual source signals s_i it is straightforward to find the state indices q_i (note that just as s_i indicates the source signal $s_i(t)$ across all time, we similarly use q_i to indicate the entire state sequence $q_i(t)$ for that source through all time); however, in the source separation problem we do not of course observe the individual sources, only their combination x.

In general, a specific parameterization of a source model provides an adequate description of a source signal; thus, if we know the appropriate state sequence q_i for source i, we may know as much detail as we care about for the corresponding signal s_i. In this case, we can re-express our original source separation problem as the problem of finding the model parameters – in this case, the best-fitting set of per-source codebook state sequences $\{\hat{q_i}\}$ instead of the best-fitting set of source signals $\{\hat{s_i}\}$. Now equation 4.4 becomes:

$$\{\hat{q_i}\} = \underset{\{q_i\}}{\text{argmax}}\ p(\{q_i\}|x, \{M_i\}) = \underset{\{q_i\}}{\text{argmax}}\ p(x|\{q_i\}) \prod_i p(q_i|M_i) \tag{4.10}$$

which due to the 'memoryless' source models can be decomposed into the product of per-time likelihoods:

$$p(x|\{q_i\}) \prod_i p(q_i|M_i) = \prod_t \left(p(x(t)|\{q_i(t)\}) \prod_i p(q_i(t)|M_i) \right) \tag{4.11}$$

This can be maximized by maximizing independently at each value of t i.e. solving for the sets of state indices $\{q_i(t)\}$ separately for every t. The 'generative' equation for observation given the unknown parameters – the analog of eqn. 4.2 – becomes:

$$p(x(t)|\{q_i\}) = \mathcal{N}(x(t); \sum_i C_i\left(q_i(t)\right), \sigma^2\mathbf{I} + \sum_i \Sigma_i) \tag{4.12}$$

Here, σ^2 is the variance due to the additive observation noise from equation 4.1, which has been extended to be a multidimensional observation by multiplying by identity matrix $\mathbf{I}$. This is added to the Σ_i terms, which provide the additional variance contributed by each of the individual sources in the mixture. Now it is clear how to express mathematically the intuitively-obvious source separation problem of figure 4.1: We can simply search across all possible combinations of the five codebook vectors of source A and the three vectors of source B to find which of those fifteen combinations maximizes eqn. 4.11, or equivalently eqn. 4.12 if the probabilities $p(q_i(t)|M_i)$ are uniform over allowable states $q_i(t)$. We do this at every time step, and gather the codebook indices for each source into the sequences plotted in the lower right of figure 4.1. We can use these index sequences to look up actual feature vector prototypes from the two codebooks to recover 'ideal' source signal sequences (i.e. the expected values, with no observation or model noise).

4.3 HIDDEN MARKOV MODELS

The derivation above was simplified by the assumption of 'memoryless' source models, so that we could separate the source signals one step at a time; the range of problems where this is sufficient to resolve sources is very limited and generally not very interesting (for instance, if the above example is interpreted as describing spectrogram slices, signal separation could be achieved with a simple fixed lowpass/highpass filter pair). However, the same approach can be applied when instantaneous observations are ambiguous – for instance when both sources are generated by the same model – provided that the models also provide adequate sequential constraints, and the observation avoids pathological ambiguities.

The most simple form of temporal constraint is a transition matrix, where the concept of the codebook index is broadened slightly to become a 'state', and the transition matrix specifies which states can occur after which other states; A realization of a signal from a model now consists of assigning a specific state to each time frame, following the transition matrix constraints. Figure 4.2 shows a new example in which both sources are using the same model (so the feature dimensions containing information about the two states inevitably overlap), but this model includes the constraint that states must move left-to-right i.e. a particular state in one time frame can only be followed in the next time frame by the same state, or its immediate neighbor to the right (wrapping around from the rightmost state back to the beginning). This is illustrated by the state diagram below the codebook vectors, and equivalently with the transition matrix, where a shaded cell indicates an allowed transition.

Although still simple, this example illustrates several important points. Looking at the codebook (now referred to as the per-state expectations or means) along with

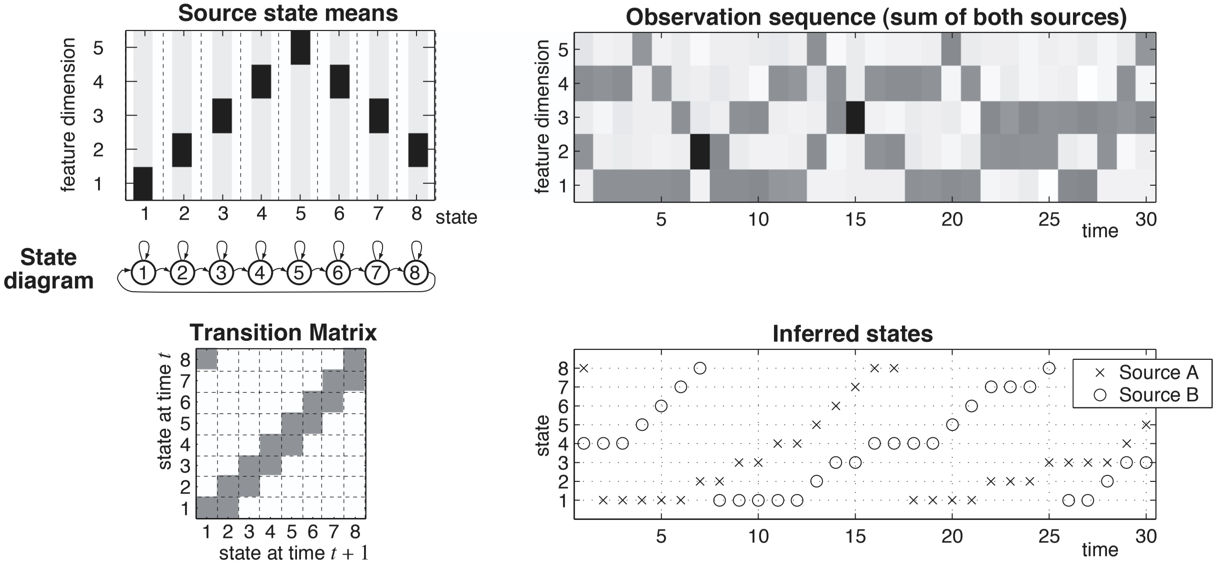


Fig. 4.2 Example of mixture composed of two sources defined by the same hidden Markov model. Sequence constraints (indicated by the equivalent State Diagram and Transition Matrix) mean that, for this observation sequence, each source can be uniquely recovered in spite of overlap between the state outputs.

the state diagram, we see that the source model again predicts a sequence of vectors, each with a single feature dimension larger than the rest. What is new is that the transition matrix constrains the sequence of states such that at any time, the next time frame can only either repeat the vector of the current frame (i.e. remain within the same state via the self-loop arc), or advance one step to the right in the state diagram to select the next vector along. This results in a pattern of the boosted dimension first moving upwards (states 1 through 5) then moving back down again (states 5 through 8), then repeating; the self-loops imply that it can take an unknown number of time frames to move on to the next state, but the 'direction of motion' of the prominent dimension will not change, except when in states 5 and 1.

The interesting consequence of this is that distinct states may have identical 'emission' characteristics (state means). Thus, states 4 and 6 both have their state mean maximum in dimension 4, and by looking at one frame alone it is impossible to distinguish between them. If, however, you had a sequence of vectors from such a source, and you looked forward or backward from a particular point in time until you saw a vector whose maximum was not in dimension 4, you can immediately distinguish between the two possible states: State 4 is always preceded by a maximum in dimension 3 and followed (after any self-loop steps) by a maximum in dimension 5; State 6 has the reverse pattern. This is the sense in which the state generalizes the codebook index: not only does it define the prototype output, but it also embodies information about what will happen in the future (and what has happened in the past).

This 'memory' encoded in the state is known as a first-order Markov property, since the behavior of the model depends on the current state but on nothing earlier in the history of the signal once the current state is known. Thus, this combination of a set of probabilistic per-state emission distributions (the codebook state means and associated variances) and a matrix of state-to-state transition probabilities is known as the hidden Markov model (HMM) [29], since the state sequence is 'hidden' by the uncertainty in the emission distributions. It turns out, however, that arbitrarily complex sequential structure can be encoded in an HMM, simply by expanding the range of values that the state can take on (the 'state space') to capture more and more detail about the prior history.

Looking at the observed mixture of two sources (top right of figure 4.2) and knowing that each source is constrained to move the prominent dimension all the way to the top then all the way to the bottom, one dimension at a time, it becomes clear that the state trajectory of the two sources is largely unambiguous (although because of the symmetry when both sources share a model, there is always a permutation ambiguity between the source labels). The bottom right pane shows both inferred state sequences, which are now plotted on the same axis since they come from the same model.

Note however that the inference is not completely unambiguous. At time step 22, source A must be in state 2 (since it was in state 1 at the previous time, but the observation is no longer consistent with state 1), and source B must be in state 7 (coming out of state 6). Similarly, by time 25, source B must be state 8 (ready to move into state 1 at step 26), so source A must be in state 3 to account for the combined observation. Since both dimensions 2 and 3 are large for all observations in-between, the state transitions of the two models must happen at the same time, but we cannot tell whether it occurs at time 23, 24, or 25 (the inference procedure has arbitrarily chosen 25). This is, however, a narrowly-bounded region of uncertainty, and soon disappears.

A more serious problem occurs if both models go into the same state at the same time. This would result in an observation similar to time step 15, with a single dimension even larger than the usual prominent dimensions. But unlike time step 15 which is explained by two different states that happen to have the same observation pattern (i.e. a collision in feature space), a collision in *state* space makes it impossible to find the correct correspondence between the discrete paths leading in to and out of the collision: both future trajectories form valid continuations from the point of state collision i.e. valid continuations of either preceding path. The best hope for an unambiguous analysis is that the two sources never arrive at the same state at the same time; clearly, this hope becomes more and more likely as the total number of states in the model becomes larger and larger, so the prior probability of any state becomes lower, and the probability of both models being in the same state at the same time drops even faster.

For the formal analysis of this situation, we note that the component source signals are still adequately described by the state index sequences (although those sequences are now subject to the constraints of the transition matrix). Our source inference problem can thus still be reduced to solving for the state sequences for each model, i.e. eqn. 4.10, repeated here:

$$\{\hat{q_i}\} = \underset{\{q_i\}}{\text{argmax}}\ p(\{q_i\}|x, \{M_i\}) = \underset{\{q_i\}}{\text{argmax}}\ p(x|\{q_i\}) \prod_i p(q_i|M_i) \tag{4.13}$$

Now, however, we cannot simply decompose this into a set of independent time slices since the transition matrix introduces dependence between time slices. We can, however, consider the time observations to be independent given the state indices, i.e.

$$p(x|\{q_i\}) \prod_i p(q_i|M_i) = \left(\prod_t p(x(t)|\{q_i(t)\}) \right) \prod_i p(q_i|M_i) \tag{4.14}$$

where $p(q_i|M_i)$ is the likelihood of an entire state sequence $\{q_i(1), q_i(2), \ldots q_i(T)\}$ under the model. We can use the chain rule to decompose this into the product of a set of conditional probabilities,

$$\begin{aligned} p(q_i|M_i) &= p(q_i(1), q_i(2), \ldots q_i(T)|M_i) \\ &= p(q_i(1)|M_i) p(q_i(2)|q_i(1), M_i) \ldots p(q_i(T)|q_i(T-1), q_i(T-2), \ldots q_i(1), M_i) \\ &= p(q_i(1)|M_i) p(q_i(2)|q_i(1), M_i) \ldots p(q_i(T)|q_i(T-1), M_i) \\ &= p(q_i(1)|M_i) \prod_{t=2}^{T} p(q_i(t)|q_i(t-1), M_i) = \prod_{t=1}^{T} p(q_i(t)|q_i(t-1), M_i) \end{aligned} \tag{4.15}$$

We hide the special case for $p(q_i(1)|M_i)$ in the final step by defining it as equal to the otherwise undefined $p(q_i(1)|q_i(0), M_i)$. The simplification of the longer conditions comes from the Markov property: Everything about the possible behavior at time $t+1$ is captured in the state at time t, so once that state appears in a conditioning expression, no earlier states need appear, or mathematically:

$$p(q_i(t+1)|q_i(t), q_i(t-1), \ldots q_i(1), M_i) = p(q_i(t+1)|q_i(t), M_i) \qquad (4.16)$$

Substituting this into eqn. 4.14, gives:

$$p(x|\{q_i\}) \prod_i p(q_i|M_i) = \prod_t \left(p(x(t)|\{q_i(t)\}) \prod_i p(q_i(t)|q_i(t-1), M_i) \right) \qquad (4.17)$$

which is the same as eqn. 4.11 from the earlier example, except for the dependence of each state index on its predecessor, $q_i(t-1)$. To find the set of state sequences that maximize this, we could almost maximize for each t independently (as above) were it not for the reference to $q_i(t-1)$. However, we can solve for the maximizing sequences using the efficient dynamic programming approach known as the Viterbi algorithm. Although it is generally described for single Markov chains, we can easily convert our problem into this form by considering the set of state indices at a particular time, $\{q_i(t)\}$ (a single state index for each source), to define the state of a super-model whose state-space is composed of the outer-product of all the component model state spaces (once again, as for the earlier example). The most-likely path through this composed state space will simultaneously solve for the complementary best paths through all the individual models. Thus, we are solving for:

$$\{\hat{q_i}\} = \underset{\{q_i\}}{\text{argmax}} \prod_t p(x(t)|\{q_i(t)\}) p(\{q_i(t)\}|\{q_i(t-1)\}, \{M_i\}) \qquad (4.18)$$

The dynamic programming trick of the Viterbi algorithm is to solve for the most-likely state sequence $\{\hat{q_i}\}$ recursively, by defining the best paths to *all* states at time $t+1$ in terms of the best paths to all states at time t, repeating until the end of the observation sequence is reached, then back-tracing the sequence of states that led to the most likely state in the final step. The point to notice is that at any given time step prior to the end of a sequence it is not possible to know which of the states will be the one on the final most-likely path, but by tracking the best paths to every one of them, we are sure to include the single one we actually want. Then, when we do get to the end of the sequence, we can quickly trace back from the state with the greatest likelihood at the final step to find all the preceding states on that best path. Thus, if $\hat{p}(\{q_i(t)\}|x)$ is the likelihood of the state sequence reaching some particular state-set $\{q_i(t)\}$ at time t that best explains x (up to that time), we can define:

$$\hat{p}(\{q_i(t)\}|x) = p(x(t)|\{q_i(t)\}) \max_{\{q_i(t-1)\}} \hat{p}(\{q_i(t-1)\}|x) p(\{q_i(t)\}|\{q_i(t-1)\}) \qquad (4.19)$$

i.e. the likelihood of the most likely state path to any particular state-set at time t is the local likelihood of the current observation given this state-set, $p(x(t)|\{q_i(t)\})$ multiplied by the best likelihood to a state-set at the preceding timestep $\hat{p}(\{q_i(t-1)\}|x)$, scaled by the transition probability between those state-sets $p(\{q_i(t)\}|\{q_i(t-1)\})$, where all state-sets $\{q_i(t-1)\}$ at the preceding timestep are searched to find the largest possible product. The best preceding state-set is saved for every state at every timestep to allow reconstruction of the best path once the most likely state-set at the end of the sequence has been found.

Thus, instead of trying to search over all possible state sequences for the best one (of order N^T calculations, for N states over T timesteps), the Markov property permits the Viterbi algorithm to keep track of only one path per model state, then incrementally work through the timesteps until the end is reached (order $T \times N^2$ calculations).

In practice, to solve the example in figure 4.2, we again construct a new state space of $8 \times 8 = 64$ source state means, corresponding to the expected outputs for every combination of the 8 states of each of two models, and a $64 \times 64 = 4096$ element transition matrix, indicating all the allowable transitions between each of the 64 super-states; since the transition matrices of the individual model states are so sparse in this example – only two out of eight allowable successors to any state – the super-state transition matrix is correspondingly sparse with only four out of 64 allowable successors from any state. (The Matlab code to reproduce this example is available at `http://www.casabook.org/`.) Having reconstructed our problem as a single HMM, we can then use a standard implementation of the Viterbi algorithm to solve for the best state sequence.

In doing so, we notice one pitfall of this approach. The combined state space can grow quite large, particularly for more than two sources, as its size is the product of the sizes of the state spaces of all the individual source models. Moreover, the transition matrix for this composed model is the square of this larger number of states, which of course gets larger more quickly. In the absence of special measures to take advantage of whatever sparsity may exist in this matrix, simply representing the transition matrix may be the limiting factor in direct implementations of this kind of system.

It should be noted that this idea of inferring multiple Markov chains in parallel is precisely what was proposed in Varga and Moore [42] as HMM decomposition, and by Gales and Young [12] as parallel model combination, both in the specific context of speech recognition – where the HMM models for speech already exist. The idea has had limited practical impact not only because of the tractability problems arising from the product state space, but also because observations of mixtures of signals can interfere with feature normalization schemes. In particular, it is normal in speech models to ignore the absolute energy of the signal (e.g. the zero'th cepstral coefficient), and focus instead on the relative distribution of energy in frequency and time. Even the spectral variation may be modeled only relative to some long-term average via feature normalizations such as cepstral mean subtraction (CMS, where every cepstral dimension is made zero-mean within each utterance prior to recognition [34]). However, to employ the kind of factorial HMM processing we have described above, it is necessary to model explicitly the absolute levels of the sources in order to predict their combined effect i.e. the spectrum of two sources combined depends critically on their energy relative to each other, something generally not available from speech recognition models. And a normalization like CMS will in general require different offsets for each of the combined source signals, meaning that CMS-based source models cannot be directly combined, since they are modeling different domains. Although these feature normalization tricks are secondary to the main pattern recognition approach of speech recognition, they are critical to its effectiveness, since without them the distributions to be modeled become far less compact and regular, requiring many more parameters and much larger training sets for equivalent performance.

4.4 ASPECTS OF MODEL-BASED SYSTEMS

For the purposes of illustration, the preceding section focused on the particular instance of hidden Markov models for sources. However, as mentioned in the introduction, any source separation must employ some kind of constraints, and constraints on the form of individual sources (instead of the relationships between sources) can be expressed and discussed as source models. Thus, the model-based perspective is very general and can be made to fit very many source separation systems. Each system is differentiated by its approach to handling a number of key questions, namely: which kinds of constraints are to be expressed (including the domain in which the model is defined), and how to specify or learn those constraints (specifying the structure of actual sources); how to fit the model to the signal (i.e. how to search the model space); and how to use the inferred model parameters to generate an output such as a resynthesized isolated source signal. We now consider each of these questions in turn.

4.4.1 Constraints: Types and Representations

The essence of model-based separation is that some prior knowledge about the expected form of the sources has to be extracted: These are the constraints that make an otherwise underconstrained model solvable, and we consider the model to be whatever it is that captures these constraints. There are two major classes of model constraints that fall under this description. The first, as used in the examples of sections 4.2 and 4.3 may be caricatured as "memorization": in some domain, with some degree of accuracy, the model strives to memorize all possible realizations of the signal class, and then the fit of a proposed signal to a model, $p(s_i|M_i)$, is evaluated by seeing if s_i matches one of the memorized instances, or falls into the set of signals that was inferred as their generalization. We will call these models 'explicit', since the representation, such as the state means in the earlier examples, are generally directly expressed in the domain of the signal or some close relative.

The alternative to this is 'implicit' models, where the subset of allowable signals is defined indirectly through some function evaluated on the signal. This is the case for many well-known computational auditory scene analysis (CASA) systems (such as Brown and Cooke [6]) which enforce an attribute such as periodicity of their outputs not by matching signals against a dictionary of known periodic signals, but by defining an algorithm (like harmonic grouping) that ensures that the only possible outputs of the system will exhibit this attribute. Such models have been discussed in chapter 3.

We now consider in more detail the different kinds of constraints, and how they are defined or learned, for each of these two classes in turn.

Explicit signal models The kind of signal model that is most easily understood is one in which, like our initial examples, instances or prototypes of the signal realizations are directly stored. An extreme example of such an explicit signal model would be if the waveform of the source signal is known and fixed, and only its time of occurrence in the mixture was unknown; in this case, a matched filter would be sufficient to estimate these times. The state-based model examples of sections 4.2 and 4.3 are slightly more general, in that the signal model is the concatenation of a sequence of predefined vectors (or distributions within the space of feature vectors), and the source signal can be composed of these 'dictionary elements' in an unspecified (but constrained) order.

Given enough states along with the transition probabilities, Markov models of this kind can in theory describe any kind of signal – in the limit by memorizing every possible individual sequence with its own set of states, although this is infeasible for interesting sound sources. Typically, the compromise is to use a dictionary that attempts to represent all possible signals with a minimum average approximation error, which will reduce as the size of the dictionary increases. As an example, the models in Roweis [35] use 8000 states (implying 64 million transition arcs), but the kinds of signals that can be constructed by sampling from the space described by the model (i.e. making random transitions and emissions according to the model distributions), while identifiably approximations of speech, are very far from fully capturing the source constraints. Nix et al. [26] go so far as to use 100,000 codebook entries; this large codebook forces them to use a stochastic search (particle filtering) to identify high-likelihood matches.

Magnitude-spectral values over windows long enough to span multiple pitch cycles (i.e. narrowband spectrogram slices, with windows in the range 20-30 ms for speech) are the most popular domain for these kinds of models because in such a representation a waveform with a static, pseudoperiodic 'sound' (say, a sung vowel) will have a more-or-less constant representation; it is also broadly true that two waveforms that differ only in phase are perceived as the same, at least as long as the phase difference between them does not change too rapidly with frequency. The disadvantage of magnitude spectra is that they do not combine linearly (although see the discussion of max-approximation in section below). Linearity can be recaptured by working directly in the waveform domain (e.g. with codebooks of short fixed-length segments of waveforms). The disadvantage is that a periodic signal will not result in a sequence of identical states, because in general the fixed-length windows will have different alignments with the cycle in successive frames, and hence significantly different waveforms. An approach to this is to construct a codebook consisting of *sets* of codewords, where each set consists of a single waveshape realized at every possible temporal alignment relative to the analysis frames [4].

One way of thinking about these models is that they are defining a subspace containing the signals associated with the modeled source, within the space of all possible sounds. Learning a dictionary of specific feature vectors is covering this subspace with a set of discrete blobs (which permits us to enumerate allowable transitions between them), but it might also be described other ways, for instance by basis vectors. Jang and Lee [19] use Independent Component Analysis (ICA) to learn statistically efficient basis functions for the waveforms of different sources, then perform separation by finding the waveforms that project onto these basis sets so as to jointly maximize the likelihood of the models constituted by the ICA-defined subspaces; their system achieves modest signal-to-noise ratio (SNR) improvements of 3-6 dB even without modeling any kind of temporal structure - each 8 ms frame is processed independently. Approaches of this kind could be based on many novel ideas in subspace learning, also known as nonlinear dimensionality reduction, such as Weinberger et al. [43].

Another approach to handling the impractically-large sets of memorized states that a dictionary needs to represent if it is to provide an accurate, detailed signal model is to look for a factorization, so that a large number of effective memorized states are actually represented as the combinations of a smaller number of state factors. One way to do this is to break a large spectral vector into a number of separate subbands, spanning different frequency ranges [32]. While this allows N prototypes for each of M subbands to represent N^M complete vectors, many of these will not correspond to observed (or likely) source spectra (unless the source's subbands are independent, which would be very unusual). This can be ameliorated to some extent through a coupled HMM, where a given state depends not only on its immediate predecessor in the same subband (a conventional HMM) but also on the states in the two flanking channels, allowing local influence to propagate all the way across the spectrum. A variational approximation can be used to decouple the bands, in the context of a particular set of states, and thus the same efficient Baum-Welch training [29] conventionally used for speech recognition HMMs can be used for each chain independently, repeated iteratively until all states converge.

A different way to factorize the prototype spectra is to model them as a combination of coarse and fine spectral structure. In speech, coarse structure arises primarily from the resonances of the vocal tract (the “formants”), and the fine structure comes from the excitation of the vocal folds (for “voiced” speech, which has an identifiable pitch) or from noise generated at constrictions in the vocal tract (for unvoiced speech such as fricatives). Factoring spectrogram frames into these two components might offer substantial gains in efficiency if it can be done in such a way that the excitation and resonance state choices are close to being independent – which seems plausible given their separate physical origins, but remains to be further investigated [21, 20].

Large, detailed, explicit signal models cannot be constructed by hand, of course. They are possible only because of the availability of effective machine learning algorithms for generalizing ensembles of training examples into parametric models such as Gaussian mixtures (for independent values) and hidden Markov models (for sequences). Although finding the optimal solution to representing a large collection of feature vectors with a limited number of prototypes in order to minimize distortion is NP-hard [13], simple, greedy algorithms can be very successful. For instance, the standard algorithm for building a vector quantization codebook, where the objective is to define a finite set of vectors that can be used to represent a larger ensemble with the smallest average squared error, is due to Linde, Buzo, and Gray [16]: for an N entry codebook, first choose N vectors at random from the ensemble as the initial codebook. Next, quantize the entire training ensemble by assigning them to the closest codeword. Then rewrite each codeword as the mean of all the vectors that were assigned to it; repeat until the improvement in average distortion of the revised codebook ceases to improve.

Implicit signal models Traditionally, CASA has referred to systems that attempt to duplicate the abilities of listeners to organize mixtures into separately-perceived sources by using computational analogs of the organizational cues described by psychologists such as Bregman [5] or Darwin and Carlyon [8]. These systems, discussed in detail in chapter 2, may be considered as falling into our model-based perspective since they extract signals by finding components within the mixture that match a restricted set of possible signals - the $p(s_i)$ of eqn. 4.4. For example, many CASA systems make heavy use of the "harmonicity cue" – the inference that a set of simultaneous but distinct sinusoids, or subbands exhibiting periodic amplitude modulation, belong to a single common source if their frequencies or modulations are consistent with a common fundamental period [44, 6, 17]. One popular way to evaluate this is by autocorrelation: energy in any frequency band that arises from a common fundamental period will exhibit a large peak at that period in its autocorrelation. Evaluating the autocorrelation in multiple subbands and looking for a large peak at a common period is one way to calculate a model fit that could easily take the form of $p(s_i|M)$. This fits into the framework of eqn. 4.4 in that candidate sources lacking the required harmonic structure are not considered i.e. assigned a prior probability of zero. (Further enhancements to include CASA cues such as common onset or spatial location can similarly be expressed as priors over possible solutions.) Such a model could of course equivalently be 'memorized' – represented in explicit form – by training a vector quantizer or Gaussian mixture model on a large number of the signals that scored highly on that metric. A sufficiently large model of this kind could be an arbitrarily accurate representation of the model, but it seems an inefficient and inelegant way to capture this simple property.

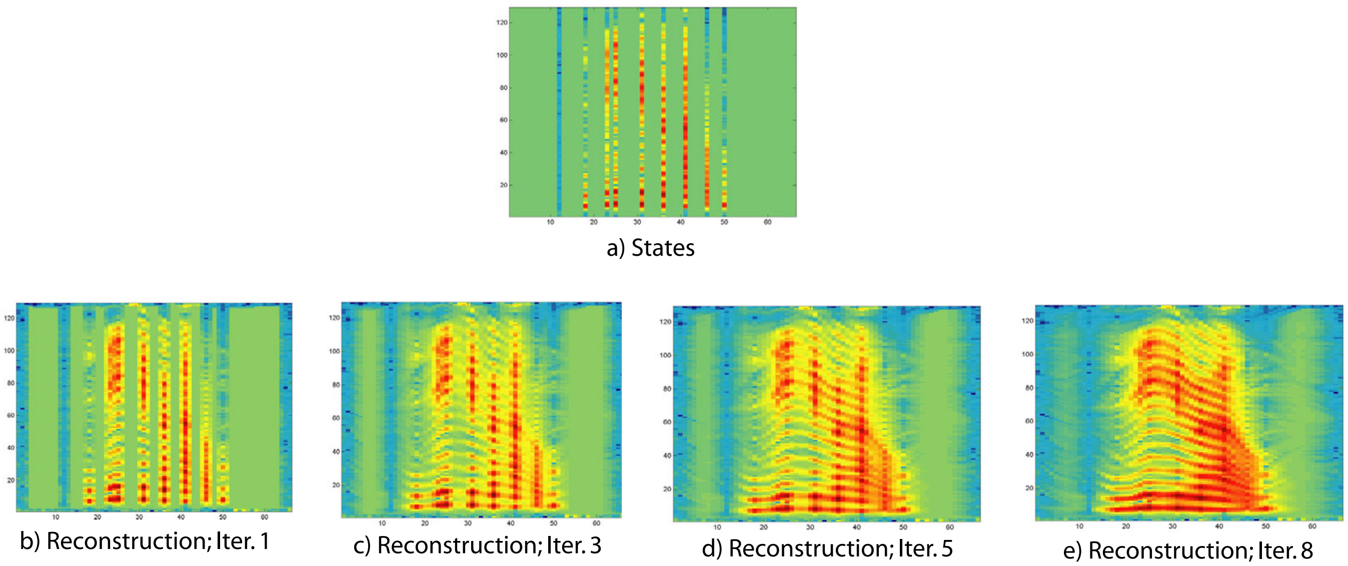
In an effort to broaden the scope of CASA beyond harmonic (pitched) signals, the system of Ellis [10] uses a hand-designed set of 'atomic elements', parametric sound fragments able to provide pitched or unpitched energy with constrained forms (such as separable time and frequency envelopes, or constant energy decay slopes) that were based on observations of real-world sources and insights from human perception. The ambiguity of decomposing mixtures into these elements led to a complex blackboard system tracking and extending multiple alternative source-set hypotheses, pruned on the basis of their 'quality' of explanation, which can be interpreted as incorporating both the match of signal to explanation $p(x|\{s_i\})$ and the match of the explanatory sources to the underlying signal models $p(s_i|M_i)$.

Considering these implicit models as mechanisms for calculating the probability of a signal fitting the model can be helpful. For instance, although much of the structure of these models is encoded by hand, there may be ways to tune a few parameters, such as weights for mapping near-misses to intermediate probabilities, that can be set to maximize the match to a collection of training examples, just as explicit models are based more or less entirely on training examples.

The "deformable spectrogram" model is an interesting middle ground between directly-encoded rules and parametric models inferred from data [33, 31]. Based on the observation that narrowband spectrograms of speech and other sources have a high degree of continuity along time, the model attempts to describe each small patch of a spectrogram as a transformation of its immediately preceding time slice, where the transformations are predominantly translations up or down in frequency. This results in a distribution of transformation indices across time and frequency, which is smoothed via local dependencies whose form is learned from examples to maximize the likelihood of training data under the model. This part of the model is similar to conventional CASA in that a particular signal property – in this case the continuity of time-frequency energy across small frequency shifts – is captured implicitly in an algorithm that can describe only signals that have this property. But when local transformations are unable to provide a good match for a particular time frame – as might happen when the properties of a sound change abruptly as a new source appears – the model has a separate 'matching' mode, where a new spectral slice is drawn from a dictionary, very much like the basic state-based models. The difference is that these memorized states are used only occasionally when tracking the deformations fails; figure 4.3 illustrates the reconstruction of a segment of speech starting from a few sampled spectra, with the intervening frames inferred by propagating these spectra using the inferred transformation maps. This recourse to states is also key to the use of this model for signal separation, since, when applied in subbands, it marks potential boundaries between different states, very much like the elemental time-frequency regions used in earlier CASA systems. Source 'separation' consists of attempting to group together the regions that appear to come from a common source, for instance by similarities in pitch, or consistency with another model better able to capture structure at longer timescales such as the pronunciation and language models of a speech recognizer.

4.4.2 Fitting models

As we have noted already, the problem with eqn. 4.4 is that it requires searching for the maximum over an exponentially-sized space of all possible combinations of all possible source signals $\{s_i\}$ – completely impractical for direct implementation. By assigning zero probability to many candidate source waveforms, source models can at least restrict the range of sets $\{s_i\}$ that needs to be considered, but exhaustive search is rarely practical. In our toy examples, we limited the temporal dependence of our source signals to at most one step into the past. This meant that we could use dynamic



a) States

b) Reconstruction; Iter. 1

c) Reconstruction; Iter. 3

d) Reconstruction; Iter. 5

e) Reconstruction; Iter. 8

Fig. 4.3 The Deformable Spectrogram matching-tracking model. A spectrogram is primarily modeled as a field of local transformations (not shown) describing how each part-slice is related to its immediate predecessors via transformation, but the system also has a dictionary of seed states that can be inserted at points when local transformation fails to describe the observations. These panes show how a signal is reconstructed from this description, starting with the discrete states (left panel), through successive iterations of propagating those slices out into adjacent frames according to the transformation fields, through belief propagation (from Reyes-Gómez [31]).

programming to effectively search over all possible sequences by considering only all possible sequential *pairs* of states at each time step – a complexity of $(N^M)^2 \times T$ for N states per source and M sources over T time steps, instead of $(N^M)^T$ for the full search over all sequences. But even with this considerable help, merely searching across all combinations of states from two or more models can quickly become unwieldy for models with useful amounts of detail. For instance, Roweis [35] had 4000 states in each model, so a full combined statespace would involve 16,000,000 codewords – approaching the practical limit for current computers – and considering all possible adjacent pairs of these combined states would involve $(4000^2)^2 = 2.56 \times 10^{14}$ likelihood calculations per time step – several weeks of calculation using today's computers – even ignoring the storage issues. Clearly, some shortcuts are required, both for representing the joint state space, and for calculating the state sequence hypotheses (i.e. 'pruning' or abandoning unpromising hypotheses early in their development).

One way to improve the efficiency of the large joint state space is to avoid pre-calculating all N^M state distributions, but create them on the fly from the individual models. With aggressive hypothesis pruning, it may be that many state combinations are never even considered and thus their expected templates need never be calculated. Since the expected observation for a combination of sources in some particular state is simply the superposition of the observations from each individual source (e.g. sum in the amplitude domain of eqn. 4.1), we can exploit properties of natural sounds to represent the observation templates implicitly by reference to the per-source states, and to prune away large chunks of the candidate state space without careful search. The approximation discussed in the next section can make this particularly efficient.

The Max-Approximation When matching spectral magnitudes (i.e. spectrogram slices), it is commonly assumed that one or other signal dominates in each cell, i.e. $|A + B| \approx max(|A|, |B|)$ at each point on the time-frequency distribution [36]. This is justified by the wide local variation in energy across even small local regions of source spectrograms, so that even if two sources have similar energy at the coarse scale, at a fine scale they will have comparable energy in only a small proportion of cells; in other cells, the magnitude ratio between the two sources is so great that the smaller can be safely ignored, and the magnitude of the combination is approximated as the magnitude of the larger component. Simple experiments show that, for randomly-selected mixtures of two voices with equal energy (recorded in quiet, for example from the TIMIT database [14]), for STFT window lengths anywhere between 2 and 100 ms, the difference in dB-magnitude between individual time-frequency cells is approximately normal distributed with a standard deviation of about 20 dB (bottom left panel of figure 4.4). Adding two cells with unknown phase alignment but a difference in magnitude of 6.5 dB results in a cell with an expected magnitude equal to the maximum of the two, but a standard deviation of 3 dB; this deviation naturally grows smaller as the difference in magnitude gets larger. Thus, only 25% of cells in a spectrogram of the sum of two clean speech signals (i.e. the proportion of normal distribution lying within $6.5 \div 20$ deviations from the mean) will have a magnitude distributed with a standard deviation greater than 3 dB away from the maximum of the two mixed components. Empirically, mixtures of two TIMIT signals have log-spectral magnitudes which differ by more than 3 dB from the maximum of the two components in fewer than 20% of cells, and the average dB difference between the max over the sources and the mixture spectrogram is very close to zero – although it is a heavy-tailed distribution, and significantly skewed, as shown in the bottom-right panel of figure 4.4. Note that the skew between mixture spectrogram and max of individual sources is negative, meaning that the mix log-spectrogram has a wider range of values when it is smaller than the sources - because two equal-magnitude sources in theory completely cancel to create a mixture cell with an unbounded negative value on a logarithmic scale; the most by which the log-magnitude of the sum of two sources can exceed the larger of their magnitudes is 6 dB.

This max-approximation can be used to accelerate the search for well-matching pairs of states [35, 36]. On the assumption that the magnitude-spectrum of an observed mixture will have energy that is (at least) the element-wise maximum of the spectra of the two components, we can sort each model's codebook by the total magnitude in each codeword in excess of the combined observation we are trying to match – related to the 'bounded integration' that can be used to match spectral models in the presence of masking, as described in chapter 9. Assuming that the max-approximation holds, codewords with elements that are larger than the corresponding mixture observation dimension cannot be a part of the mixture, or else the magnitudes in those observation elements would be larger. Thus, all code words from both models containing dimensions that significantly exceed the observation can be eliminated, and only combinations of the remaining states need be considered. Further, when searching for states to pair with a given state to complete an explanation, dimensions in which the first state is significantly below the observation can be used to further restrict the set of states from the second model to be considered, since the responsibility for explaining these so-far unaccounted dimensions will fall fully on the second state. Once the codebooks have been so pruned, the actual best match can be found as the closest match under the max-approximation; it would even be possible to use a more sophisticated function to combine each candidate pair of states into an expected distribution for the joint distributions, since only a few of these state pairs need be considered.

Of course, the dynamic programming search relies not only on the match between the state's emission distribution and the actual observation $(p(x(t))|\{q_i(t)\})$ from eqn. 4.19, but also the transition costs from all the preceding states being considered $(p(\{q_i(t)\}|\{q_i(t-1)\}))$; in theory, a wide variation in transition costs could completely reorder the likelihood of different states to be very different from an ordering based on emission match alone, although in practice many observation (mis)matches constitute such small probabilities that they will never be boosted back into viability through large transition likelihoods. In any case, estimating and storing the N^2 values of the full transition matrix for each model may be difficult; one alternative is to ignore transition probabilities altogether, but learn codebooks for overlapping windows of features (e.g. pairs of adjacent frames), and let the obligation to match the

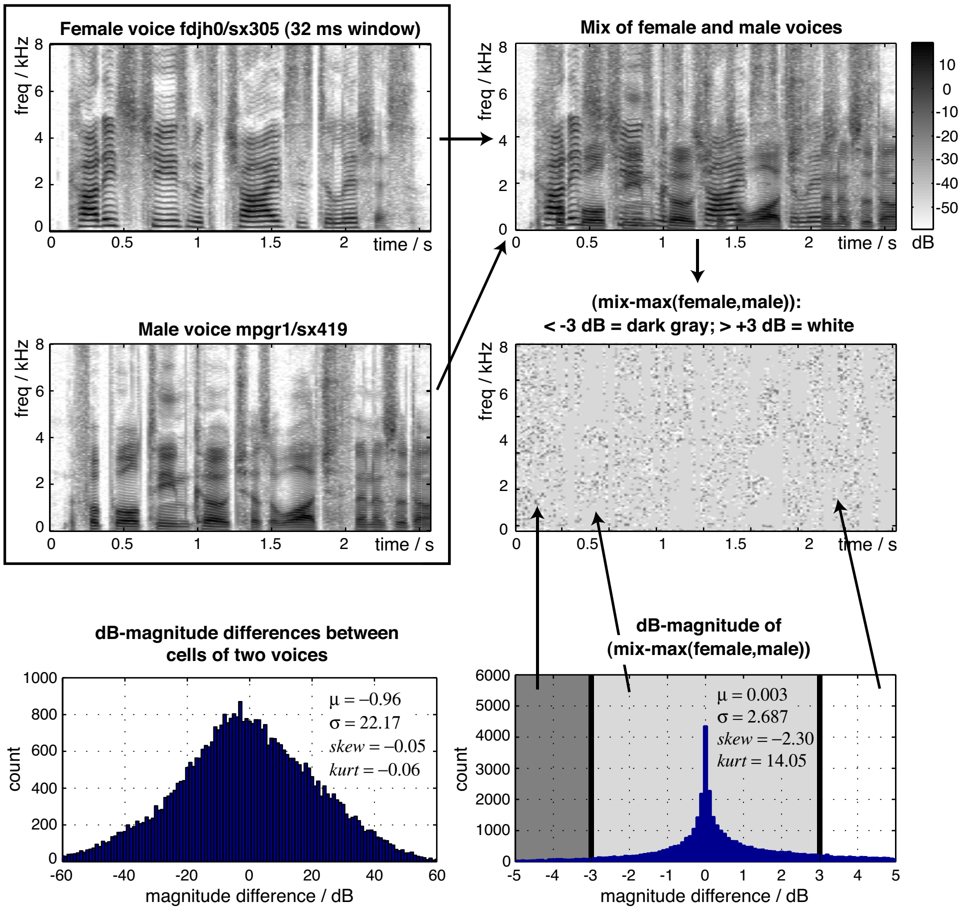


Fig. 4.4 Illustration of the relationship between the magnitudes of time-frequency distributions of two voices. Top left: Spectrograms of female and male voices; top right: spectrogram of the sum of the two waveforms. Middle right: time-frequency cells in which the log-magnitude of the sum is more than 3 dB different from the maximum of the log-magnitudes of the individual sources (18% of cells in this example). Bottom left: histogram showing the distribution of log-magnitude differences between corresponding cells in the two source spectrograms. Bottom right: histogram of the differences between the spectrogram of the mixed signal and the maximum of the two component spectrograms; the region between the solid lines corresponds to the neutral-colored cells in the middle-right panel.

overlapping frame shared between two states introduce constraints on which states can follow others [36].

When the source magnitudes are comparable in particular dimensions (e.g. closer than 6.5 dB), the max assumption involves a fairly significant departure from reality, but it is possible to make a more accurate inference of the true underlying per-source magnitudes. Kristjansson et al. [22] use the "Algonquin" algorithm (iterative linearization of a more complex relationship) to infer source spectra that achieve about 6 dB reduction of interferer energy for separating single channel mixtures of male and female voices based on 512 component Gaussian mixtures and 128 dimension spectrogram frames, for a multiple-speaker, limited-vocabulary corpus (TIDIGITS [24]).

In cases where hypothesis probabilities can be evaluated, but there is no efficient algorithm for searching systematically through an intractable hypothesis space, there are also stochastic search methods that move around locally in the hypothesis space searching for the best local optimum that can be found; clearly, such approaches are very dependent on their initialization, and on the complexity of the hypothesis likelihood function. Nix et al. [26] use Sequential Monte Carlo (also known as 'particle filtering') to develop a fixed-size set of many thousands of individual samples in the hypothesis space; at each step, the least likely hypothesis 'particles' are eliminated, and the most likely ones are duplicated, to evolve along slightly different random paths within those higher-likelihood areas. Although the algorithm can be computationally expensive (e.g. thousands of times slower than realtime), it offers a relatively straightforward approach to searching intractable spaces.

These codebook models draw techniques and inspiration from the large hidden Markov models used in contemporary large vocabulary speech recognizers, which frequently contain thousands of states and sometimes even hundreds of thousands of Gaussian components. Speech recognizers, however, have the advantage that, since just one source is present, the absolute level of the signal can be normalized, or simply ignored in modeling (for instance by excluding the zero'th cepstral coefficient from models, or by normalizing feature dimensions along time within an utterance, which can also remove any static spectral coloration). When two or more sources are combined, however, their relative amplitude (and fixed spectral shaping) has a central impact on the resulting waveform, so individual source models must include these factors, either directly within the state set, or as separate factors that must somehow be inferred and tracked for each source during inference.

Search in implicit models When we cast traditional CASA models such as Brown and Cooke [6] or Hu and Wang [17] into the model search of eqn. 4.4, it draws attention to the search over alternative candidate source signals which is generally described as grouping in those models, and constructed in such a way as to minimize the number of alternatives that are explicitly considered. In both those systems, a source is defined by a pitch track extracted from the mixture, then all time-frequency cells 'consistent' with that pitch track are assigned to the source mask. Because these models do not attempt to estimate the probabilities of their mask assignments, it would be difficult to compare and choose between alternative hypotheses.

Other models go beyond this. In the 'speech fragment decoder' of Barker et al. [2], separating speech from interference is cast as finding the best subset to cover the target speech from among a collection of relatively large time-frequency regions (that might arise from a CASA-like initial segmentation, although most of their results come from a cruder breakup of regions poking out of a static noise-floor estimate). They do this by incorporating the search for the best 'segregation hypothesis' (set of regions assigned to the target speech) into the existing speech recognizer search across phones and words for the best-fitting model. Since the outcome of this search is the pair of utterance and segregation mask that afford the greatest likelihood between observed mixture and the recognizer's speech models (exploiting the missing-data matching of Cooke et al. [7]), the speech models are directly guiding the search by rejecting any segregation masks that imply target energy that is inconsistent with the utterance hypotheses being considered. This is a very neat way to combine the heuristic segmentation principles of CASA systems with the highly-developed and powerful models of speech signals contained within current speech recognizers; however, these models actually do not attempt to enforce or exploit consistency in the character of the target voice because they are speaker-independent i.e. deliberately trained on a range of speakers to accept any 'typical speech' segments equally well. These issues are developed further in chapter 9.

By contrast, Shao and Wang [38] use models drawn from contemporary work in speaker identification (instead of speech recognition) to organize smaller fragments. Given a set of segments extracted from a mixture of two voices by looking for partial pitch tracks, they identify both speakers by finding the two models that together best account for all the segments (from an ensemble of previously-trained single-speaker models), and can then assign individual segments to each speaker based on likelihood, thereby segregating these partial representations of the sources. The speaker models, which are typically large Gaussian mixture distributions for the instantaneous spectra produced by each speaker, do not provide or encode any temporal sequence constraints or structure, which, while presumably making them weaker, also makes them much simpler to apply.

4.4.3 Generating output

By approaching a problem as blind signal separation, the implication is that the desired outputs are the actual original waveforms comprising the mixture. But when a problem is described as scene analysis, it is no longer so clear what the desired outputs should be: If we are attempting to duplicate human perceptual organization, there is no strong reason to believe that the brain ever constructs a representation close to the raw waveform of the individually-perceived sources; perhaps a much more abstract description of distinct sources in terms of high-level attributes is sufficient. Such alternatives become increasingly clear in the context of model-based scene analysis, since models also represent waveforms in terms of a smaller set of more abstract parameters such as state paths, pitch tracks, or word-string hypotheses.

Of course, there are situations in which extracting waveforms is relatively easy – predominantly when multiple channels are available (e.g. from multiple microphones). In this case there may exist combinations of time-invariant filters through which the channels may be combined to largely or completely cancel out interfering sources, although this generally relies on there being only a small number of spatially-compact sources of interfering noise. Although this scenario, considered in more detail in chapter 6, is amenable to a range of more general source separation algorithms such as beamforming and independent component analysis (ICA) [3, 18], there are opportunities for models to help still further. Seltzer et al. [37] obtain a target for gradient-descent optimization of the coefficients of their filter-and-sum microphone array by looking at the means of the HMM states from the best path of a conventional speech recognizer applied to the array output. As the speech signal is enhanced through incrementally refined filter coefficients, the recognizer alignment is re-estimated to achieve iterative improvement of both speech recognizer result and filter coefficients. The system enhances the speech against nonspeech interference, and also ensures that the filter-and-sum operation results in a speech signal that is spectrally balanced to resemble the recognizer's training set. Reyes-Gómez et al. [30] use a similar approach to separate two simultaneous speakers, using a variational approximation to solve the factorial HMM problem of identifying the separate state paths for the two speakers; this approach is highly relevant to the model-based analysis presented in this chapter, and readers are directed to Ghahramani and Jordan [15].

It is, however, the single channel case that this chapter is mostly concerned with. In that case, only trivial mixtures (those with non-overlapping spectral support) can be separated via linear time-invariant filtering. If, however, we allow the filters to vary with time, we can still extract multiple, distinct waveforms by assigning the same spectral region to different sources at different times. This is the essence of time-frequency masking, where the magnitudes of the cells of an invertible time-frequency representation (i.e. a short-time Fourier transform in which the phase information is retained) are scaled by element-wise multiplication with a time-frequency mask. Masks are often binary, meaning that some cells are entirely deleted (reduced to zero magnitude), and some are left unchanged (multiplied by 1). Time-frequency masking has been used in the majority of single-channel signal separation systems, including Brown and Cooke [6], Roweis [35] and Hu and Wang [17], and can give startlingly good results, particularly when the mask is chosen 'optimally' to pass only cells where the local energy of target exceeds interference. From Wiener filtering, we know that optimizing the SNR (i.e. minimizing the total energy of any distortion relative to the original signal) is achieved by scaling the mixture of signal-plus-noise by $1/(1 + P_N/P_S)$ where P_N and P_S are the noise and signal power respectively. Applying this to each time-frequency cell individually leads to a 'soft mask' that optimizes the overall SNR by maximizing the contribution of each cell. Quantizing this optimal mask to the closest binary level (1 or 0) gives the best binary mask, which deletes all cells in which $1/(1+P_N/P_S) < 1/2$ i.e. $P_N > P_S$, or a local SNR below 0 dB. This is in line with our expectations, since each included cell increases the noise denominator by the amount of interference included in the cell at the same time as reducing it by the target energy that is saved from deletion; total noise is minimized by including only cells where the net noise contribution is negative i.e. target exceeds interference. Time-frequency masking can be viewed as the combination of a set of time-varying filters, one for each frequency channel, whose frequency response is fixed but whose gain switches dynamically between zero and one as successive time frames in that frequency band are passed or excluded. The limitation of time-frequency masking is that it cannot do anything substantial to separate target and interference energy that fall into a single cell. Even if ‘soft masks’ are used (scaling cells by a real value in an effort to reconstruct the magnitude of one source even when a cell is significantly influenced by both), scaling only the magnitude will not give a particularly high-quality reconstruction of the target source which would rely on recovering the source, not mixture, phase as well.

As we have explained, CASA systems usually operate by identifying larger time-frequency regions and assigning them to one of several inferred sources. Thus, the core representation of these sources is a binary time-frequency map that can be used directly for time-frequency masking. For explicit models, such as those that calculate the sequence of codewords that are inferred to best approximate the source, there are several options for resynthesis. Usually the model states do not include phase information, so cannot be directly resynthesized to a waveform. (Magnitude-only reconstruction procedures have been proposed [25], but they rely on consistency between overlapping windows, which is likely to be seriously disrupted in the quantized magnitude sequences derived from state-based models [28, sec. 7.5].)

One source for the missing phase information needed for conventional STFT overlap-add inversion is the original mixture: taking the state-based magnitude inferences and copying the phase from the time-frequency analysis of the original mixture will give some kind of reconstruction in which interference is reduced. However, cells in the original mixture which are not dominated by the target will have inappropriate phase, and the energy reconstructed from those cells is more likely to be perceived as interference: it would be better to eliminate those cells altogether, as in time-frequency masking. Roweis [35] converts his state-based inferences for the two component sources into a binary mask simply by comparing the codewords from the two models, and including in the mask only those cells where the inferred target magnitude is larger than the inferred interference. After time-frequency masking, the magnitude of the resynthesis in these cells may not precisely match the inferred magnitude vectors (since both phase and magnitude are being taken from the mixture), but because they arise from the same underlying signal, the result is likely to have less distortion: any magnitude quantization in the codebook now has no direct impact – all the codebooks have to do is correctly model the relative magnitudes of target and interference in each cell; the exact levels are less important.

This approach, of using possibly crude estimates of multiple components to choose a time-frequency mask, could be used with any system that estimates the spectral magnitude of both source and interference, but more complex, parametric models have at least the possibility of recovering the signal even in time-frequency regions dominated by interference, provided the model carries enough information for resynthesis. Thus, the ‘weft’ elements of Ellis [11] modeled periodic energy with a single pitch and a smooth time-frequency envelope that could be interpolated across occlusions; this is sufficient parametric information to resynthesize a signal that, while different from the original in the fine detail of its phase structure, can carry much the same perceptual impression.

In many cases, however, it may be that the abstract, parametric internal representation is a useful and appropriate output from the scene analysis. Thus, although the system of Ellis [10] analyzed mixtures into elements that could be individually resynthesized and recombined into waveform approximations, the higher-level description of the mixture in terms of these parametric elements would be more useful for many applications, including classifying the contents of the mixture, or searching for particular events within the audio content. Similarly, the 'speech fragment decoder' of Barker et al. [2] performs an analysis of a source mixture, but its primary output is the word-string from the speech recognizer it contains. Outputs other than waveform have major implications for the evaluation of scene analysis systems since there is no longer any possibility of using an SNR-like measure of success; however, given that these systems are often working at a level abstracted away from the raw waveform, SNR is frequently an unsatisfying and misleading metric, and the effort to devise measures better matched to the task and processing would be well spent [9].

4.5 DISCUSSION

So far we have presented the basic ideas behind model-based scene analysis, and examined the common issues that arise in constructing systems that follow this approach – representing the constraints, searching the hypothesis space, and constructing suitable outputs. In this section we discuss some further aspects of model based separation and its relation to other blind signal separation approaches such as ICA and sparse decomposition.

4.5.1 Unknown interference

So far our analysis has been symmetric in all the sources that make up a mixture. Although the goal of our processing may be to find out about a particular target source, the procedure involved estimating *all* the sources $\{s_i\}$ in order to check their consistency with the mixed observations x. This is often seen as a weakness. For instance, in speech enhancement applications, although we are very interested in our target signal, and can often expect to have the detailed description of it implied by the model $p(s_i|M_i)$, we don't really care at all about the interference signal, and we don't know (or want to assume) very much about it. Requiring a detailed, generative model for this interference is either very limiting, if our enhancement will now only work when the interference matches our noise model, or very burdensome, if we somehow attempt to build a detailed model of a wide range of possible interference.

This raises a point that we have skirted over thus far, namely how the match to the observation constrains the complete set of source signal hypotheses. Eqn. 4.4 searches over the product $p(x|\{s_i\}) \prod_i p(s_i|M_i)$, and we have focused mainly on how this is limited by the model match likelihoods $p(s_i|M_i)$, but of course it will also reject any hypotheses where $p(x|\{s_i\})$ is small i.e. the set of proposed sources would not combine to give something consistent with the observations. So another natural limitation on the search over $\{s_i\}$ is that once all but one of the sources has a hypothesis the last remaining source is strongly constrained by the need to match the observation. Returning to the simple linear combination model of eqns. 4.1 and

4.2, this could be a tighter constraint that the source model i.e. the full distribution over the N^{th} source given the observation and the hypotheses for the other sources becomes:

$$\begin{aligned} p(s_N|x, \{s_1 \ldots s_{N-1}\}) &\propto p(x_i|\{s_i\})p(s_N|M_N) \\ &= \mathcal{N}(x; \sum_i s_i, \sigma^2)p(s_N|M_N) \\ &= \mathcal{N}(s_N; x - \sum_{i \backslash N} s_i, \sigma^2)p(s_N|M_N) \end{aligned} \tag{4.20}$$

Either the first term (the normal distribution arising from the model of the domain combination physics) or the second term (our familiar source model) can provide tight constraints, and one approach to finding a s_N is simply to take the mode of the normal distribution, $\hat{s}_N = x - \sum_{i \backslash N} \hat{s}_i$ (i.e. $x - \hat{s}_0$ in the simplest case of a single target with hypothesized value $\hat{s}_0$ and a single interfering source) and multiply its likelihood by $p(\hat{s}_N|M_N)$; if the model distribution is approximately constant over the scale of variation represented by σ^2, this single point will suffice to evaluate the total likelihood of any set of sources involving $\{s_1 \ldots s_{N-1}\}$.

The point to remember when considering the problem of unknown interference is that although we have focused on source models that provide tight constraints on the source components, the analysis still applies for much looser models of $p(s_i|M_i)$ able to accept a broad range of signals. In particular, if the hypothesized mix contains only one such 'loose' model (for instance in the mix of a single well-characterized voice and single 'garbage model' of everything else), then the source inference search is still straightforward when the $p(x|\{s_i\})$ constraint is applied.

The more specific a background noise model can be, the more discrimination it adds to the source separation problem. A simple, stationary noise model that posits a large variance in every dimension may give a similar likelihood to background noise with or without an added foreground spectrum, since the changes due to the foreground result in a spectrum that is still well within the spread covered by the noise model. A more sophisticated noise model, however, might be able to capture local structure such as slow variation in noise power; a running estimate of the current noise level would then allow a much tighter model of the expected noise at each frame, and better discrimination between target states embedded in that noise. Thus, even though the specific spectral form of interference may be unknown in advance, there may be other constraints or assumptions that can still be captured in a kind of model, to allow separation. By the same token, target-extraction systems that do not have an explicit source model for the interference may often be equivalent (or approximations) to a model-based analysis with a particular, implicit model for noise.

4.5.2 Ambiguity and adaptation

When a signal is being analyzed as the combination of a number of sources without any prior distinction between the properties of each source (such as a mixture of otherwise unspecified voices), there is of course an ambiguity over which source will appear under each of the system's source indices. As long as correspondence remains fixed throughout the signal, this is unlikely to cause a problem. However, if models really are identical, there may be cases where it is possible to lose track of the sources and permute the source-to-output arrangement mid stream – for instance if two speakers both happen to fall silent at the same time, without any difference in context to constrain which of the continuations belongs with which history. This is the problem of state-space collision as mentioned in section 4.3. In the case of two speakers, the problem could be resolved by noting that within the speaker-independent space covered by the generic models, each particular source was falling into a particular subspace, determined by the particular characteristics of each speaker; by updating the source models to incorporate that observation, the symmetry could be broken and the ambiguity resolved. This process bears strong similarities to model adaptation in speech recognition, where a generic model is altered (for instance, through an affine transformation of its codeword centers) to achieve a better fit to the signal so far observed [23]. In model-based separation, where separation power derives from the specificity of match between model and source, there is great benefit to be had from online tuning of certain model parameters such as gain, spectral balance, spatial location etc. that may vary independently of the core source signal and are thus amenable to being described via separate factors.

4.5.3 Relations to other separation approaches

CASA is frequently contrasted with Independent Component Analysis [3, 18] as a very different approach to separating sources that is based on complex and heuristic perceptual models rather than simple, intrinsic properties; at the same time, Smaragdis [39] has argued that the psychoacoustic principles of Auditory Scene Analysis are more compactly explained simply as aspects of more fundamental properties such as the independence of sources. Unlike ICA, the model-based approach we have described *assumes* the independence of the source signals (by multiplying together their individual probabilities $p(s_i|M_i)$), whereas ICA would calculate and attempt to maximize some measure of this independence. The approaches can, however, be complementary: model parameters may provide an alternative domain in which to pursue independence [19], and prior source models can help with problems such as correctly permuting the independent sources found in separate frequency bands in spectral-domain convolutive ICA [40].

Another closely-related approach is sparse decomposition [46, 27], where a signal or mixture is expressed as the sum of a set of scaled basis functions drawn from an overcomplete dictionary – that is, a dictionary with more entries than there are degrees of freedom in the domain. Because the dictionary is overcomplete, there are many combinations of basis functions that will match the observation with equal accuracy, so additional objectives must be specified. By finding the solution with the minimum total absolute magnitude of basis scaling coefficients, the solution is driven to be sparse, with the majority of values close to zero i.e. the signal has been modeled by combining the smallest number of dictionary elements. When the dictionary adequately covers the sources, and the sources have distinct bases, the source separation is very high quality. This is in a sense a generalization of the large memorization models discussed in this chapter, since, for linear combination, if the basis sets consist of the codewords and the signal is a match to one of them, then sparse decomposition will find the appropriate codeword and the decomposition will be a single nonzero coefficient. In addition, because the coefficients are scalars, sparse decomposition automatically encompasses all scaled versions of its basis functions, as well as linear combinations i.e. the entire subspace spanned by its bases. At the same time, specification of the overcomplete dictionaries is a critical part of the problem with no single solution, and scaled and/or combined instances of the codewords may not be appropriate to the problem. Searching for the sparse solution is also a major computational challenge.

Given that CASA is motivated by the goal of reproducing the source organization achieved by listeners, we may ask if model-based scene analysis bears any relation to perceptual organization. In his seminal account of sound organization in listeners, Bregman [5] draws a sharp distinction between *primitive* and *schema-based* auditory scene analysis. *Primitive* processes include the fusion of simultaneous harmonics and energy sharing a common onset: because these organizing principles reflect the intrinsic physics of sound-producing mechanisms in the world, Bregman argues that these abilities come ready-made in the auditory system, since, based on the engineering received wisdom of his day, he sees it as needlessly inefficient to require each individual to learn (acquire) these regularities from the environment when they could be pre-programmed by evolution. *Schema-based* organization, by contrast, is explicitly based on an individual's specific, recent and/or repeated experiences, and is needed to explain preferential abilities to segregate, for instance, voices in a particular language, as well as the way that certain segregation abilities can improve with practice or effort – not the kind of properties one would expect from low-level pre-programmed neural hardware.

While the model-based approach may seem like a natural fit to Bregman's schema, some caution is advised. Firstly, the kinds of perceptual phenomena that schema are used to explain, such as learning of specific large-scale patterns like a particular melody or a new language, are not particularly addressed by the techniques described above, which are still mainly focused at low-level moment-to-moment signal separation. Secondly, we have argued that model-learning approaches can actually do away with the need for separate hard-wired *primitive* mechanisms, since these regular patterns could be learned relatively easily from the environment. In the 15 years since Bregman's book was published, there has been a definite shift in favor of engineering systems that learn from data rather than coming pre-programmed with explicit rules. While some level of environment-adapted hardware specialization is obviously needed, it is at least plausible to argue that evolution could have left the majority of real-world regularities to be acquired by the individual (rather than hard-coding them), finding this to be the most efficient balance between functionality and efficiency of genetic encoding. While there is strong evidence for a distinction in listeners between how harmonics are fused into tones, versus higher-level phenomena such as how phonemes are fused into words, it is interesting to note that these could both be the result of making inferences based on patterns learned from the environment, where there might be a spectrum of timescales and prior confidences associated with different classes of learned patterns, yet each could constitute an instance of a single basic memorization process.

4.6 CONCLUSIONS

Source models are an effective and satisfying way to express the additional constraints needed to achieve single-channel source separation or scene analysis. Many approaches can be cast into the hypothesis search framework, and the probabilistic perspective can be a powerful domain for integrating different information and constraints. Major problems remain, however: simplistic modeling schemes, such as attempting to memorize every possible short time frame that a source can emit, are inadequate for the rich variety seen in real-world sources such as human speech, and any rich model presents an enormous search space that cannot really be covered except with special, domain-specific tricks to prune hypotheses under consideration. Both these areas (better models and more efficient search) present major areas for future research in model-based scene analysis, and in particular can benefit from innovations in signal description and analysis described elsewhere in this volume.

REFERENCES

1. F. R. Bach and M. I. Jordan. Blind one-microphone speech separation: A spectral learning approach_ In *Advances in Neural Information Processing Systems (NIPS)*, volume 17. MIT Press, Cambridge MA, 2004. URL `http://cmm.ensmp.fr/~bach/fbach_nips_2004_speech.pdf`.

2. J. Barker, M. P. Cooke, and D. P. W. Ellis. Decoding speech in the presence of other sources. *Speech Communication*, 42:5–25, 2005. URL `http://www.ee.columbia.edu/~dpwe/pubs/BarkCE05-sfd.pdf`.

3. A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995. URL `ftp://ftp.cnl.salk.edu/pub/tony/bell.blind.ps.Z`.

4. T. Blumensath and M. Davies. Unsupervised learning of sparse and shift-invariant decompositions of polyphonic music. In *Proc. IEEE Int. Conf. Acous., Speech, and Sig. Proc.*, pages V–497–500, Montreal, 2004.

5. A. S. Bregman. *Auditory Scene Analysis*. Bradford Books, MIT Press, 1990.

6. G. J. Brown and M. P. Cooke. Computational auditory scene analysis. *Computer speech and language*, 8:297–336, 1994.

7. M. P. Cooke, P. D. Green, L. B. Josifovski, and A. Vizinho. Robust automatic speech recognition with missing and unreliable acoustic data. *Speech Communication*, 34(3):267–285, 2001.

8. C. J. Darwin and R. P. Carlyon. Auditory grouping. In B. C. J. Moore, editor, *The Handbook of Perception and Cognition, Vol 6, Hearing*, pages 387–424. Academic Press, 1995.

9. D. P. W. Ellis. Evaluating speech separation systems. In P. Divenyi, editor, *Speech Separation by Humans and Machines*, chapter 20, pages 295–304. Kluwer, 2004. URL `http://www.ee.columbia.edu/~dpwe/pubs/Ellis04-sepeval.pdf`.

10. D. P. W. Ellis. *Prediction–driven computational auditory scene analysis*. PhD thesis, Department of Electrtical Engineering and Computer Science, M.I.T., 1996. URL `http://web.media.mit.edu/~dpwe/pdcasa/pdcasa.pdf`.

11. D. P. W. Ellis. The weft: A representation for periodic sounds. In *Proc. IEEE Int. Conf. Acous., Speech, and Sig. Proc.*, pages II–1307–1310, 1997. URL `http://www.ee.columbia.edu/~dpwe/pubs/icassp97-wefts.pdf`.

12. M. J. F. Gales and S. J. Young. Hmm recognition in noise using parallel model combination. In *Proc. Eurospeech-93*, volume 2, pages 837–840, 1993.

13. M. R. Garey, D. S. Johnson, and H. S. Witsenhausen. The complexity of the generalized lloyd-max problem. *IEEE Transactions on Information Theory*, 28 (2), 1982.

14. J. Garofolo. Getting started with the darpa timit cdrom: An acoustic phonetic continuous speech database, 1993. URL `http://www.ldc.upenn.edu/Catalog/LDC93S1.html`.

15. Z. Ghahramani and M.I. Jordan. Factorial hidden markov models. *Machine Learning*, 1997.

16. R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, 1984.

17. G. Hu and D.L. Wang. Monaural speech segregation based on pitch tracking and amplitude modulation. *IEEE Transactions on Neural Networks*, 15(5), September 2004.

18. A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 1999.

19. G.-J. Jang and T.-W. Lee. A probabilistic approach to single channel blind signal separation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1173–1180, 2002. URL `http://books.nips.cc/papers/files/nips15/SP02.pdf`.

20. T. Kristjansson, 2003. Personal communication.

21. T. Kristjansson and J. Hershey. High resolution signal reconstruction. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding ASRU*, 2003. URL `http://mplab.ucsd.edu/~jhershey/publications/ASRU_paper.pdf`.

22. T. Kristjansson, H. Attias, and J. Hershey. Single microphone source separation using high resolution signal reconstruction. In *Proc. IEEE Int. Conf. Acous., Speech, and Sig. Proc.*, pages II–817–820, Montreal, 2004. URL `http://research.goldenmetallic.com/icassp04_1mic.pdf`.

23. C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9:171–186, 1995.

24. R. G. Leonard. A database for speaker independent digit recognition. In *Proc. IEEE Int. Conf. Acous., Speech, and Sig. Proc.*, pages III–42–45, 1984.

25. S. H. Nawab, T. F. Quatieri, and J. S. Lim. Signal reconstruction from short-time fourier transform magnitude. *IEEE Trans. on Acous., Speech, and Sig. Proc.*, 31 (4):986–998, 1983.

26. J. Nix, M. Kleinschmidt, and V. Hohmann. Computational auditory scene analysis by using statistics of high-dimensional speech dynamics and sound source direction. In *Proc. Eurospeech*, pages 1441–1444, Geneva, 2003. URL `http://www.physik.uni-oldenburg.de/Docs/medi/members/michael/papers/Nix_Eurospeech_2003.pdf`.

27. Barak A. Pearlmutter and Anthony M. Zador. Monaural source separation using spectral cues. In *Proc. Fifth International Conference on Independent Component Analysis ICA-2004*, 2004. URL `http://www-bcl.cs.may.ie/~bap/papers/HRTF-ICA2004.pdf`.

28. T. F. Quatieri. *Discrete-time Speech Signal Processing: Principles and Practice*. Prentice Hall, Upper Saddle River, NJ, 2002. ISBN 0-13-242942-X.

29. L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proc. IEEE*, 77(2):257–286, Feb 1989. URL `http://www.ee.columbia.edu/~dpwe/e6820/papers/Rabiner89-hmm.pdf`.

30. M. Reyes-Gómez, B. Raj, and D. P. W. Ellis. Multi-channel source separation by beamforming trained with factorial hmms. In *Proc. IEEE Workshop on Apps. of Sig. Proc. to Audio and Acoustics*, pages 13–16, Mohonk NY, October 2003. URL `http://www.ee.columbia.edu/~dpwe/pubs/waspaa03-muchan.pdf`.

31. M. J. Reyes-Gómez. *Statistical Graphical Models for Scene Analysis, Source Separation and Other Audio Applications*. PhD thesis, Department of Electrical Engineering, Columbia University, New York, NY, 2005.

32. M. J. Reyes-Gómez, D. P. W. Ellis, and N. Jojic. Multiband audio modeling for single channel acoustic source separation. In *Proc. IEEE Int. Conf. Acous., Speech, and Sig. Proc.*, Montreal, 2004. URL `http://www.ee.columbia.edu/~dpwe/pubs/icassp04-muband.pdf`.

33. M. J. Reyes-Gómez, N. Jojic, and D. P. W. Ellis. Deformable spectrograms. In *Proc. AI and Statistics*, Barbados, 2005. URL http://www.ee.columbia.edu/~dpwe/pubs/aistats05-defspec.pdf.

34. A. E. Rosenberg, C.-H. Lee, and F. K. Soong. Cepstral channel normalization techniques for HMM-based speaker verification. In *Int. Conf. on Speech and Lang. Proc.*, pages 1835–1838, Yokohama, September 1994.

35. S. Roweis. One-microphone source separation. In *Advances in NIPS 11*, pages 609–616. MIT Press, Cambridge MA, 2001.

36. S. Roweis. Factorial models and refiltering for speech separation and denoising. In *Proc. EuroSpeech*, Geneva, 2003. URL http://www.cs.toronto.edu/~roweis/papers/eurospeech03.pdf.

37. M. Seltzer, B. Raj, and R.M Stern. Speech recognizer-based microphone array processing for robust hands-free speech recognition. In *Proc. IEEE Int. Conf. Acous., Speech, and Sig. Proc.*, pages I–897–900, Orlando, 2002.

38. Y. Shao and D.L. Wang. Model-based sequential organization in cochannel speech. *IEEE Transactions on Speech and Audio Processing*, 2005.

39. P. Smaragdis. Exploiting redundancy to construct listening systems. In Pierre Divenyi, editor, *Speech Separation by Humans and Machines*. Kluwer, 2004.

40. P. Smaragdis. Blind separation of convolved mixtures in the frequency domain. In *Proc. Int. Workshop on Indep. & Artif. Neural Networks*, Tenerife, 1998. URL http://sound.media.mit.edu/~paris/paris-iann98.ps.gz.

41. P. Smaragdis. Non-negative matrix factor deconvolution: Extraction of multiple sound sources from monophonic inputs. In *Proc. International Congress on Independent Component Analysis and Blind Signal Separation ICA-2004*, Granada, Spain, September 2004. URL http://www.merl.com/publications/TR2004-104/.

42. A. Varga and R. Moore. Hidden markov model decomposition of speech and noise. In *Proc. IEEE Int. Conf. Acous., Speech, and Sig. Proc.*, pages 845–848, 1990.

43. K. Q. Weinberger, B. D. Packer, and L. K. Saul. Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 381–388, Barbados, Jan 2005. URL http://www.cis.upenn.edu/~lsaul/papers/kmf_aistats05.pdf.

44. M. Weintraub. *A theory and computational model of auditory monoaural sound separation*. PhD thesis, Department of Electrical Engineering, Stanford University, 1985.

45. O. Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847, July 2004. URL `http://ee.ucd.ie/~srickard/YilmazRickard2004.pdf`.

46. M. Zibulevsky and B. A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation*, 13(4):863–882, April 2001. URL `http://www-bcl.cs.may.ie/~bap/papers/nc-spica.pdf`.