ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language xxx (2009) xxx–xxx

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Super-human multi-talker speech recognition: A graphical modeling approach

John R. Hershey [a,*], Steven J. Rennie [a], Peder A. Olsen [a], Trausti T. Kristjansson [b]

[a] *IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA*
[b] *Google New York, 75 Ninth Avenue, New York, NY 10011, USA*

Received 13 August 2007; received in revised form 2 June 2008; accepted 10 November 2008

Abstract

We present a system that can separate and recognize the simultaneous speech of two people recorded in a single channel. Applied to the monaural speech separation and recognition challenge, the system out-performed all other participants – *including human listeners* – with an overall recognition error rate of 21.6%, compared to the human error rate of 22.3%. The system consists of a speaker recognizer, a model-based speech separation module, and a speech recognizer. For the separation models we explored a range of speech models that incorporate different levels of constraints on temporal dynamics to help infer the source speech signals. The system achieves its best performance when the model of temporal dynamics closely captures the grammatical constraints of the task. For inference, we compare a 2-D Viterbi algorithm and two loopy belief-propagation algorithms. We show how belief-propagation reduces the complexity of temporal inference from exponential to linear in the number of sources and the size of the language model. The best belief-propagation method results in nearly the same recognition error rate as exact inference.

Keywords: Factorial hidden Markov model; Speech separation; Algonquin; Multiple talker speaker identification; Speaker-dependent labeling

1. Introduction

One of the hallmarks of human perception is our ability to solve the auditory cocktail party problem: we can direct our attention to a given speaker in the presence of interfering speech, and understand what was said remarkably well. The same cannot be said for conventional automatic speech recognition systems, for which interfering speech is extremely detrimental to performance.

There are several ways to approach the problem of noise-robust speech recognition (see Cooke et al., 2009 for an overview of the current state of the art). This paper presents a model-based approach to the problem,

[*] Corresponding author. Tel.: +1 914 945 1814.
E-mail addresses: jrhershe@us.ibm.com (J.R. Hershey), sjrennie@us.ibm.com (S.J. Rennie), pederao@us.ibm.com (P.A. Olsen), trausti@google.com (T.T. Kristjansson).

0885-2308/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2008.11.001

Table 1
Overall word error rates across all conditions on the challenge task. *Human*: average human error rate; *IBM*: our best result; *Next best*: the best of the other published results on the challenge task Virtanen (2006); *No processing*: our recognizer without any separation; and *Chance*: the theoretical error rate for random guessing.

System	Human (%)	IBM (%)	Next best (%)	No processing (%)	Chance (%)
Word error rate	22.3	21.6	34.2	68.2	93.0

which utilizes models of both the target speech, the acoustic background, and the interaction between the acoustic sources to do robust speech separation and recognition. The system addresses the *monaural speech separation and recognition challenge* task (Cooke et al., 2009), and outperforms all other results published to date on this task. The goal of speech separation challenge task (Section 2) is to recognize the speech of a target speaker, using single-channel data that is corrupted by an interfering talker. Both speakers are constrained to be from a closed speaker set, and conform to a specific grammar. An interesting aspect of the challenge is that the organizers have conducted listening experiments to characterize human recognition performance on the task. The overall recognition performance of our system *exceeds* that of humans on on the challenge data (see Table 1).

The system is composed of three components: a *speaker recognizer*, a *separation system*, and a single-talker *speech recognizer*, as shown in Fig. 1. The core component of the system is the model-based separation module. The separation system is based upon a factorial hidden Markov model (HMM) that incorporates multiple layers of constraints on the temporal dynamics of the sources. Single-channel speech separation has previously been attempted using Gaussian mixture models (GMMs) on individual frames of acoustic features. However such models tend to perform well only when speakers are of different gender or have rather different voices (Kristjansson et al., 2004).

When speakers have similar voices, speaker-dependent mixture models cannot unambiguously identify the component speakers. In such cases it is helpful to model the statistical dependencies across time. Several models in the literature have done so for single-talker recognition (Varga and Moore, 1990; Gales and Young, 1996) or enhancement (Ephraim, 1992; Roweis, 2003). Such models have typically been based on a discrete-state hidden Markov model (HMM) operating on a frame-based acoustic feature vector.

Conventional speech recognition systems use a smoothed log spectrum in which the effects of the harmonic structure of the voice are removed. This harmonic structure is associated with the pitch of the voice, whereas only the formant structures, retained in the smoothed log spectrum, are thought to be relevant to speech recognition in non-tonal languages. However, with multiple speakers, voices tend to largely overlap and obscure each other in the smoothed log spectrum. In contrast, in the full (high-resolution) log spectrum, the different harmonic structures of the two voices greatly decreases the amount of overlap. Our separation system thus operates directly on the log spectrum. A caveat is that modeling the dynamics of speech in the log spectrum is challenging in that different components of speech, such as the pitch and the formant structure of the voice, evolve at different time-scales.

We address the issue of dynamics by testing four different levels of dynamic constraints in our separation model: *no dynamics*, low-level *acoustic dynamics*, high-level *grammar dynamics*, and a layered combination, *dual dynamics*, of the acoustic and grammar dynamics. The acoustic dynamics constrain the short-term dynamics of the pitch and formant structure together, whereas the grammar constrains the dynamics of the formant structure. In the dual dynamics condition the acoustic dynamics are intended to make up for the lack of constraints on the pitch. In the experiments we have conducted to date, grammar-level dynamics are necessary to achieve the best results. However, we have not yet seen a significant benefit of the additional acoustic constraints in the dual-dynamics model.

The acoustic models of the speakers in the system are combined to model the observed speech mixtures using two different methods: a nonlinear model known as *Algonquin* (Kristjansson et al., 2004), which models the combination of log-spectrum models as a sum in the power spectrum, and a simpler *max model* that combines two log spectra using the max function.

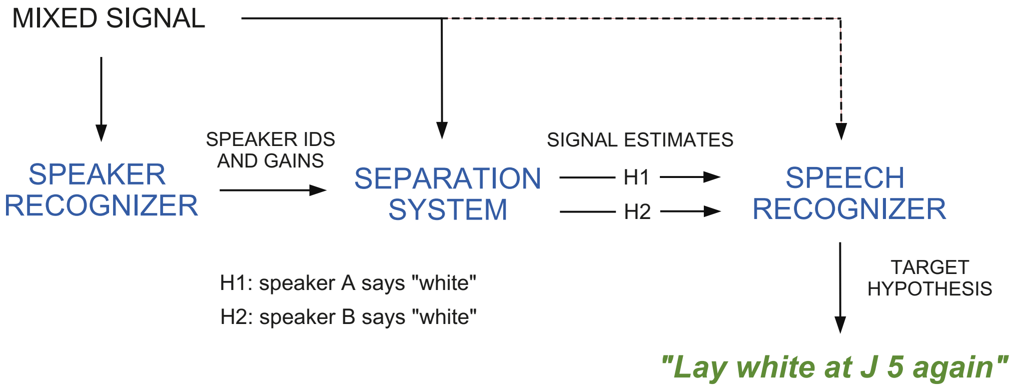


Fig. 1. System overview. The speaker recognizer (Section 7), first estimates the speaker identities and gains of both talkers. The separation system (Section 3) combines the task grammar with speaker-dependent acoustic models and an acoustic interaction model to estimate two sets of sources; one set based on the hypothesis that speaker a is the target (H1), the other on the hypothesis that speaker b is the target (H2). The single-talker speech recognition system then recognizes each signal using *speaker-dependent labeling* (Section 8) and outputs the target decoding result that yields the highest likelihood.

For a given separation model topology and inference algorithm, Algonquin and the max model perform comparably in terms of word recognition rate (WER) on the challenge: within 1% absolute overall. However the max model is considerably faster and can be run exactly, whereas Algonquin is too slow to run without the use of optimization tricks. Among other tricks, we employed band quantization (Linde et al., 1980; Bocchieri, 1993), which reduced the number of operations required to compute the acoustic likelihoods by several orders of magnitude.

Exact inference in the separation model can be done efficiently by doing a Viterbi search over the joint state space of the sources that exploits the factorial structure of the model. This approach, however, still scales exponentially with acoustic and language model size, and therefore cannot be easily be applied to larger problems.

In this paper, we additionally present a new iterative algorithm for approximate inference that employs the loopy belief propagation method to make inference scale linearly with language model size. The WER performance of this approach closely matches that of our joint inference algorithm at a fraction of the computational cost. When the identities and gains of the sources are known, the loopy belief propagation method still surpasses the average performance of humans on the task.

The performance of our best system, which does joint inference and exploits the grammar constraints of the task, is remarkable: the system is often able to accurately extract two utterances from a mixture even when they are from the same speaker.[1] Overall results on the challenge are given in Table 1, which shows that our closest competitors are human listeners.[2]

This paper is organized as follows. Section 2 describes the speech separation task. Sections 3 and 4 describe the speech models and speech interaction models used by our separation system. Section 5 describes how temporal inference can be done efficiently using a factorial Viterbi search, and in linear time using loopy belief propagation. Section 6 describes how band quantization, joint-state pruning, and an approximate max model can be used to efficiently compute the joint acoustic likelihoods of the speakers. Section 7 describes the multi-talker speaker recognition component of the system, which is based upon a simple expectation–maximization (EM) algorithm that exploits the temporal sparsity of speech to identify multiple speakers in linear time. Section 8 describes the speech recognition component of our system, which incorporates SDL: a speaker-adaptation strategy that actually performs better than using oracle knowledge of the target speaker's identity.

[1] Demos and information can be found at: http://www.research.ibm.com/speechseparation/.

[2] The different speaker conditions receive unequal weights, due to the distribution of the test set. With equal weights, the humans achieve a task error rate of 21.8%, compared to 20.6% for our system.

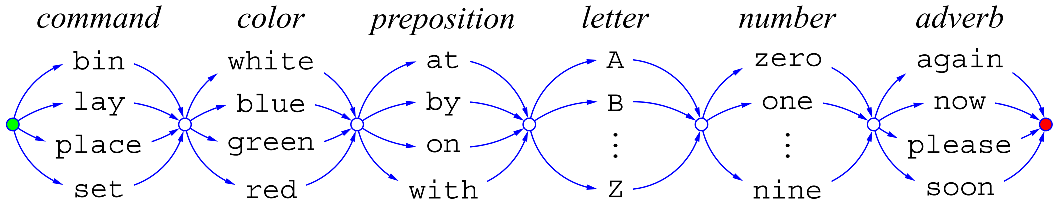


Fig. 2. Task grammar. Note that the letter *W* is excluded. An example sentence would be “lay white with *G* six please”.

Section 9 characterizes the performance of our system as a function of separation model topology and inference algorithm. We also discuss new experiments that vary the task constraints to determine the effect of different aspects of the task on performance. In order to remove the effect of the speaker recognizer, oracle speaker IDs and gains were used for these experiments. The WER performance of our best system with oracle speaker IDs and gains is 19.0%. We present results for the following cases, (a) using gender-dependent models instead of speaker-dependent background models: 23.1%; (b) when the background grammar is an unordered *bag of words*: 23.2%; (c) when the transcripts of the speakers are known: 7.6%, and (d) when the speech signals are iteratively inferred using a loopy belief propagation algorithm: 22.1%.

2. The speech separation and recognition challenge

The main task in the monaural speech separation and recognition challenge is to recognize the speech of a target speaker in the presence of another, masking speaker, using a single microphone. The speech data are drawn from the recently-collected GRID corpus (Cooke et al., 2006), which consists of simple sentences drawn from the grammar in Fig. 2. The specific task is to recognize the letter and digit spoken by the target speaker, where the target speaker always says “white” while the masker says “blue”, “green” or “red”.

The development and test data consists of mixtures that were generated by additively mixing target and masking utterances at a range of signal-to-noise ratios (SNRs): 6, 3, 0, -3, -6, and -9 dB.

The test set has 600 mixed signals at each SNR: 221 where the target and masker are the same speaker, 179 where the target and masker are different speakers of the same gender, and 200 where the target and masker are of different gender. The development set is similar but half the size at 300 mixtures per SNR.

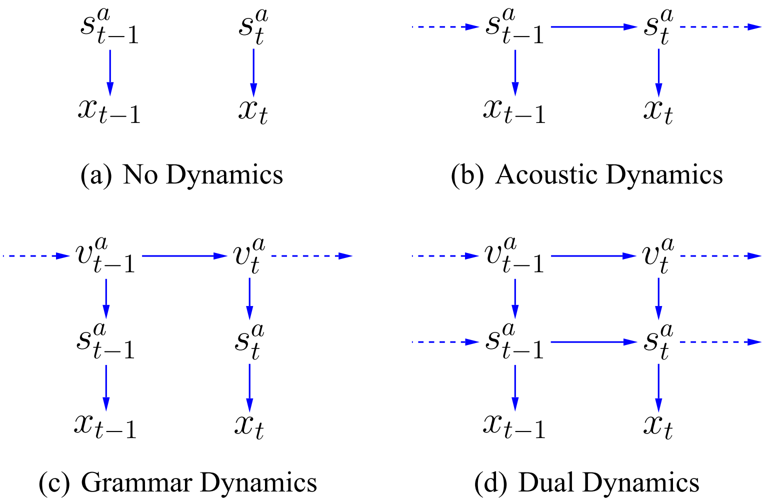
The target and masking speakers are chosen from a closed set of 34 speakers, consisting of 16 female and 18 male subjects. Clean training data, consisting of 500 utterances from each speaker, was provided for learning speech models.

The challenge also provides a stationary noise development and test set, where the task is to recognize the speech of the target speaker in the presence of “speech-shaped” stationary noise. The test data consists of mixtures that were generated by additively mixing the target utterance with speech-shaped noise at the following SNRs: 6, 0, -6, and -12 dB.

In this paper, our focus will be on the primary task of the challenge: separating speech from speech. We use the the stationary noise condition to adapt our speech recognizer.

3. Speech separation models

The separation system consists of an *acoustic model* and a *temporal dynamics model* for each speaker, as well as an *acoustic interaction model*, which describes how the source features are combined to produce the observed mixture spectrum. The acoustic features consist of short-time windowed log spectra, computed every 15 ms, to produce T frames for each test signal. Each 40 ms frame is analyzed using a 640-point mixed-radix FFT. After discarding the DC component, the log power spectrum feature vector $\mathbf{y}$ has $F = 319$ dimensions.



(a) No Dynamics (b) Acoustic Dynamics

(c) Grammar Dynamics (d) Dual Dynamics

Fig. 3. Graph of models for a given source. In (a) there are no dynamics, so the model is a simple mixture model. In (b) only acoustic dynamics are modeled. In (c) grammar dynamics are modeled, with the grammar state variables sharing the same acoustic Gaussians, in (d) dual – grammar and acoustic – dynamics have been combined. Note that (a), (b), and (c) are special cases of (d), with different assumptions of independence.

3.1. Acoustic model

For a given speaker a we model the conditional probability of the log power spectrum of each source signal $\mathbf{x}^a$ given a discrete acoustic state s^a as Gaussian, $p(\mathbf{x}^a|s^a) = N(\mathbf{x}^a; \boldsymbol{\mu}_{s^a}, \boldsymbol{\Sigma}_{s^a})$, with mean $\boldsymbol{\mu}_{s^a}$ and covariance matrix $\boldsymbol{\Sigma}_{s^a}$. For efficiency and tractability we restrict the covariance to be diagonal. This means that $p(\mathbf{x}^a|s^a) = \prod_f N(x^a_f; \mu_{f,s^a}, \sigma^2_{f,s^a})$, for frequency f. Hereafter we drop the f when it is clear from context that we are referring to a single frequency. We used $D_s = 256$ Gaussians to model the acoustic space of each speaker. A model with no dynamics can be formulated by producing state probabilities $p(s^a)$, and is depicted in Fig. 3a.

3.2. Acoustic dynamics

To capture the low-level dynamics of the acoustic signal, we model the acoustic dynamics of a given speaker, a, via state transitions $p(s^a_t|s^a_{t-1})$ as shown in Fig. 3. There are 256 acoustic states, hence for each speaker a, we estimate 256 × 256 transition probabilities.

3.3. Grammar dynamics

We use a dictionary of pronunciations to map from the words in the task grammar to sequences of three-state context-dependent phoneme models. The states of the phonemes of each word in the grammar are uniquely identified by a *grammar state*, v^a. The entire task grammar can then be represented by a sparse matrix of state transition probabilities, $p(v^a_t|v^a_{t-1})$.

The association between the grammar state v^a and the acoustic state s^a is captured by the transition probability $p(s^a|v^a)$, for speaker a. These are learned from clean training data using inferred acoustic and grammar state sequences. The grammar state sequences are computed by alignment with the reference text, using a speech recognizer with the same set of grammar states. The acoustic state sequences are computed using the acoustic model above. The grammar of our system has 506 states, so we estimate 506 × 256 conditional probabilities.

3.4. Dual dynamics

The dual-dynamics model combines the acoustic dynamics with the grammar dynamics. In general using the full combination of s and v states in the joint transitions $p(s_t^a|s_{t-1}^a, v_t)$ would make learning and inference expensive. Instead, we approximate this as $\frac{1}{z}p(s_t^a|s_{t-1}^a)^\alpha p(s_t^a|v_t)^\beta$, where α and β adjust the relative influence of the two probabilities, and z is the normalizing constant.

Note that all of the speech separation models use a common set of acoustic Gaussians $p(\mathbf{x}^a|s^a)$. This serves two purposes. First, it allows the different architectures to be compared on an equal footing. Second, it can be more efficient than having separate GMMs for each grammar state, since we have fewer Gaussians to evaluate.

4. Acoustic interaction models

The short-time log spectrum of the mixture y_t, in a given frequency band, is related to that of the two sources x_t^a and x_t^b via the *acoustic interaction model* given by the conditional probability distribution, $p(y_t|x_t^a, x_t^b)$. We consider only interaction models that operate independently on each frequency for analytical and computational tractability. The joint distribution of the observation and source in one feature dimension, given the source states is thus:

$$p(y_t, x_t^a, x_t^b|s_t^a, s_t^b) = p(y_t|x_t^a, x_t^b)p(x_t^a|s_t^a)p(x_t^b|s_t^b). \tag{1}$$

To infer and reconstruct speech we need to compute the likelihood of the observed mixture given the acoustic states,

$$p(y_t|s_t^a, s_t^b) = \int p(y_t, x_t^a, x_t^b|s_t^a, s_t^b)\,\mathrm{d}x_t^a\mathrm{d}x_t^b, \tag{2}$$

and the posterior expected values of the sources given the acoustic states and the observed mixture,

$$E(x_t^a|y_t, s_t^a, s_t^b) = \int x_t^a p(x_t^a, x_t^b|y_t, s_t^a, s_t^b)\,\mathrm{d}x_t^a\mathrm{d}x_t^b, \tag{3}$$

and similarly for x_t^b. These quantities, combined with a prior model for the joint state sequences $\{s_{1..T}^a, s_{1..T}^b\}$, allow us to compute the minimum mean squared error (MMSE) estimators $E(\mathbf{x}_{1..T}^a|\mathbf{y}_{1..T})$ or the maximum *a posteriori* (MAP) estimate $E(\mathbf{x}_{1..T}^a|\mathbf{y}_{1..T}, \hat{s^a}_{1..T}, \hat{s^b}_{1..T})$, where $\hat{s}_{1..T}^a, \hat{s}_{1..T}^b = \arg\max_{s_{1..T}^a, s_{1..T}^b} p(s_{1..T}^a, s_{1..T}^b|\mathbf{y}_{1..T})$, and the subscript, $1..T$, refers to all frames in the signal.

The acoustic interaction model can be defined in a number of ways. We explore two popular candidates, for which the integrals in (2) and (3) can be readily computed: *Algonquin*, and the *max model*.

4.1. Algonquin

The discrete Fourier transform is linear: a linear mixture of two signals in the time domain is equivalent to a mixture, $Y = X^a + X^b$, of their complex Fourier coefficients X^a and X^b, in each frequency band. Thus in the power spectrum domain we have

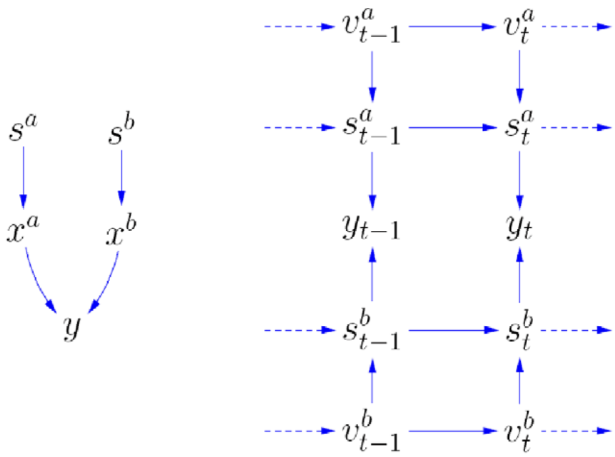
$$|Y_t|^2 = |X_t^a|^2 + |X_t^b|^2 + 2\sqrt{|X_t^a||X_t^b|}\cos\theta_t, \tag{4}$$

where θ_t is the phase angle between the two sources. When using models that ignore phase, it is reasonable to assume that the phase difference between two independent signals is uniformly distributed. The phase term above then has zero mean, and we are left with the following expected value:

$$|Y_t|^2 \approx E_\theta\left(|Y_t|^2 \;\middle|\; |X_t^a|, |X_t^b|\right) = |X_t^a|^2 + |X_t^b|^2 \tag{5}$$

Taking this approximation into the log power spectrum domain, where $x_t^a \overset{\text{def}}{=} \log|X_t^a|^2$ (and similarly for x_t^b, and y_t) we have:

$$y_t \approx \log\left(\exp(x_t^a) + \exp(x_t^b)\right) \tag{6}$$



(a) Interaction Model (b) Two speaker dual dynamics model

Fig. 4. Model combination for two-talkers. The interaction model (a) is used to link the two sources models to form (b) the full two speaker dual dynamics model, where we have simplified the graph by integrating out x^a and x^b. The other models are special cases of this graph with different edges removed, as in Fig. 3.

Algonquin models the approximation error using a Gaussian in the log domain:

$$p(y_t|x_t^a, x_t^b) = N\left(y_t; \log\left(\exp(x_t^a) + \exp(x_t^b)\right), \psi\right) \tag{7}$$

where the variance, ψ, allows for uncertainty about the phase (Kristjansson et al., 2004). An iterative Newton–Laplace method is used to linearize $\log\ (\exp(x_t^a) + \exp(x_t^b))$, and approximate the conditional posterior $p(x_t^a, x_t^b|y_t, s_t^a, s_t^b)$ as Gaussian. This allows us to analytically compute the observation likelihood $p(y_t|s_t^a, s_t^b)$ and expected value $E(x_t^a|y_t, s_t^a, s_t^b)$. The Algonquin method is well documented elsewhere, for example in Frey et al. (2001).

4.2. Max model

It was recently shown in Radfar et al. (2006) that if the phase difference between two signals is uniformly distributed, then the expected value of the log power of the sum of the signals is

$$E_\theta\left(y_t|x_t^a, x_t^b\right) = \max\left(x_t^a, x_t^b\right). \tag{8}$$

The max model uses this expected value as an approximate likelihood function,

$$p\left(y_t|x_t^a, x_t^b\right) = \delta_{y_t - \max\left(x_t^a, x_t^b\right)} \tag{9}$$

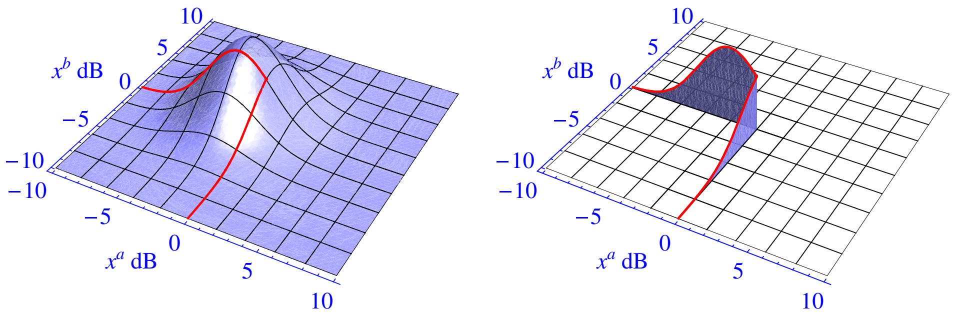
where $\delta_{(.)}$ is a Dirac delta function. The max model was originally introduced as a heuristic approximation. It was first used in Nádas et al. (1989) for noise adaptation. In Varga and Moore (1990), such a model was used to compute state likelihoods and find the optimal state sequence. In Roweis (2003), a simplified version of the max model was used to infer binary masking values for refiltering.

Here we compute feature posteriors, so that we can compute the MMSE estimators for the log power spectrum. The max model likelihood function is piece-wise linear and thus admits a closed form solution for the posterior, $p(x^a, x^b|y, s^a, s^b)$, and the necessary integrals. Fig. 5 illustrates the posterior. The likelihood of the observation given the states is

$$p\left(y_t|s_t^a, s_t^b\right) = p_{\mathrm{x}_t^a}\left(y_t|s_t^a\right)\Phi_{\mathrm{x}_t^b}\left(y_t|s_t^b\right) + p_{\mathrm{x}_t^b}\left(y_t|s_t^b\right)\Phi_{\mathrm{x}_t^a}\left(y_t|s_t^a\right), \tag{10}$$

using $p_{\mathrm{x}_t^a}(y_t|s_t^a) \overset{\mathrm{def}}{=} p(\mathrm{x}_t^a = y_t|s_t^a)$ for random variable x_t^a, and the normal cumulative distribution function $\Phi_{\mathrm{x}_t^a}(y_t|s_t^a) = \int_{-\infty}^{y_t} N(x_t^a; \mu_{s_t^a}, \sigma^2_{s_t^a})\mathrm{d}x_t^a$.

Please cite this article in press as: Hershey, J.R. et al., Super-human multi-talker speech recognition: ..., Computer Speech and Language (2009), doi:10.1016/j.csl.2008.11.001



(a) Prior density (b) Posterior density

Fig. 5. Max model: (a) the prior normal density, $p(x^a|s^a) \times p(x^b|s^b)$ is shown for a single feature dimension. Its intersection with likelihood delta function $\delta_{y-\max(x^a,x^b)}$, for $y = 0$, is represented by the red contour. (b) the likelihood, $p(y = 0|s^a, s^b)$ is the integral along this contour, and the posterior, $p(x^a, x^b|y = 0, s^a, s^b)$ is the prior evaluated on this contour, normalized to integrate to one. Marginal expected values can be easily computed from this posterior since it is composed of truncated Gaussians. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The expected value of the hidden source, given the observation and the hidden states, is

$$E\left(x_t^a|y_t, s_t^a, s_t^b\right) = \pi_a y_t + \pi_b \left(\mu_{s_t^a} - \frac{\sigma^2_{s_t^a} p_{x_t^a}(y_t|s_t^a)}{\Phi_{x_t^a}(y_t|s_t^a)} \right), \tag{11}$$

where $\pi_a = p_{x_t^a}(y_t|s_t^a)\Phi_{x_t^b}(y_t|s_t^b)/p(y_t|s_t^a, s_t^b)$ and $\pi_b = 1 - \pi_a$. Eqs. (10) and (11) were first derived in Nádas et al. (1989).

5. Temporal inference

In Hershey et al. (2006) exact inference of the state sequences (*temporal inference*), was done using factorial HMM 2-D Viterbi search, for the grammar dynamics model. Given the most likely state sequences for both speakers, MMSE estimates of the sources are computed using Algonquin or the max model. Once the log spectrum of each source is estimated, we estimate the corresponding time domain signal as described in Kristjansson et al. (2004).

The exact inference algorithm can be derived by combining the state variables, into a joint state $s_t = (s_t^a, s_t^b)$ and $v_t = (v_t^a, v_t^b)$ as shown in Fig. 6. The model can then be treated as a single hidden Markov model with transitions given by $p(v_t^a|v_{t-1}^a) \times p(v_t^b|v_{t-1}^b)$ and likelihoods from Eq. (1). However inference is more efficient if the two-dimensional Viterbi search is used to find the most likely pair of state sequences $v_{1..T}^a$, $v_{1..T}^b$. We can then perform an MMSE estimate of the sources by averaging over the posterior probability of the mixture components given the grammar Viterbi sequence, and the observations.

On the surface, the 2-D Viterbi search would seem to be of complexity $O(TD^4)$, because it requires finding for each of $D \times D$ states at the current time t the best of $D \times D$ states at the previous time $t - 1$. In fact, it can be computed in $O(TD^3)$ operations. This stems from the fact that the dynamics for each chain are independent. The forward–backward algorithm for a factorial HMM with N source models requires only $O(TND^{N+1})$ rather than the $O(TD^{2N})$ required for a naive implementation (Ghahramani and Jordan, 1995). Here we show how this can be achieved for the Viterbi algorithm.

In the 2-D Viterbi algorithm, the following recursion is used to compute, for each hypothesis of v_t^a and v_t^b, the probability of the most likely joint state sequence leading up to that pair of states, given all observations up to the previous time step:

$$q\left(v_t^a, v_t^b\right) = \max_{v_{t-1}^a, v_{t-1}^b} p\left(v_t^a|v_{t-1}^a\right) p\left(v_t^b|v_{t-1}^b\right) p\left(y_{t-1}|v_{t-1}^a, v_{t-1}^b\right) q\left(v_{t-1}^a, v_{t-1}^b\right), \tag{12}$$

where we define $q(v_1^a, v_1^b) = p(v_1^a)p(v_1^b)$. This joint maximization can be performed in two steps, in which we store the intermediate maxima:

Please cite this article in press as: Hershey, J.R. et al., Super-human multi-talker speech recognition: ..., Computer Speech and Language (2009), doi:10.1016/j.csl.2008.11.001

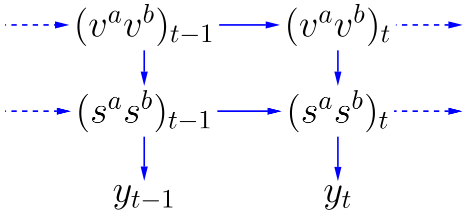


Fig. 6. Cartesian product model: Here we graph the full two-talker grammar dynamics model with acoustic and grammar states combined across talkers into single Cartesian product states, and with x^a and x^b integrated out for simplicity. In the dual dynamics model acoustic states are also connected across time, introducing cycles into the graph.

$$\tilde{q}(v^a_{t-1}, v^b_t) = \max_{v^b_{t-1}} p(v^b_t|v^b_{t-1})p(y_{t-1}|v^a_{t-1}, v^b_{t-1})q(v^a_{t-1}, v^b_{t-1}), \tag{13}$$

$$q(v^a_t, v^b_t) = \max_{v^a_{t-1}} p(v^a_t|v^a_{t-1})\tilde{q}(v^a_{t-1}, v^b_t). \tag{14}$$

We also store the value of v^b_{t-1} that maximizes (13) for each value of v^a_{t-1} and v^b_t:

$$\tilde{v}^b_{t-1}(v^a_{t-1}, v^b_t) = \arg\max_{v^b_{t-1}} p(v^b_t|v^b_{t-1})p(y_{t-1}|v^a_{t-1}, v^b_{t-1})q(v^a_{t-1}, v^b_{t-1}), \tag{15}$$

For each hypothesis of v^a_t and v^b_t, we use (14) to get the optimal value of v^a_{t-1}:

$$\hat{v}^a_{t-1}(v^a_t, v^b_t) = \arg\max_{v^a_{t-1}} p(v^a_t|v^a_{t-1})\tilde{q}(v^a_{t-1}, v^b_t). \tag{16}$$

This result is used with (15) to get the corresponding optimal value of v^b_{t-1}:

$$\hat{v}^b_{t-1}(v^a_t, v^b_t) = \tilde{v}^b_{t-1}(\hat{v}^a_{t-1}(v^a_t, v^b_t), v^b_t). \tag{17}$$

The two maximizations require $O(D^3)$ operations with $O(D^2)$ storage for each time step. This generalizes readily to the N-dimensional Viterbi search, using $O(TND^{N+1})$ operations.

A similar analysis applies to computing the grammar state likelihood $p(y|v^a, v^b, \ldots v^N)$. Naively the complexity would be $O(ND_s^N D_v^N)$ for each time step. However, we can take advantage of the factorial structure of the model, $p(s^a, s^b, \ldots, s^N|v^a, v^b \ldots v^N) = p(s^a, s^b, \ldots, s^N)p(s^a|v^a)p(s^b|v^b)\ldots p(s^N|v^N)$, to reduce this complexity to $O(\sum_{k=1}^N D_s^{N-k+1} D_v^k) \leqslant O(ND^{N+1})$, where $D = \max(D_s, D_v)$.

5.1. Belief propagation

The 2-D Viterbi search suffers from a combinatorial explosion in the number of speakers and grammar states, with complexity $O(TND_v^{N+1})$ for the grammar dynamics model. This is because it requires evaluating the joint grammar states (v^a_t, v^b_t) of the sources at each time step. Instead, we can do inference by iteratively updating the sub-models of each speaker. Using the max-product belief propagation method, Kschischang et al., 2001, temporal inference can be accomplished with complexity $O(TND_v^2)$, scaling linearly in the number of speakers. In addition to decoupling the grammar state dynamics across sources, the max-product algorithm also decouples the acoustic to grammar state interaction across sources. This reduces the complexity of the acoustic to grammar state interaction from $O(ND^{N+1})$ to $O(ND_sD_v)$ per iteration for each time step.

The max-product inference algorithm can be viewed as a generalization of the Viterbi algorithm to arbitrary graphs of random variables. For any probability model defined over a set of random variables $x \triangleq \{x_i\}$:

$$p(x) \propto \prod_{\mathcal{C}\in\mathcal{S}} f_{\mathcal{C}}(x_{\mathcal{C}}) \tag{18}$$

where the factors $f_{\mathcal{C}}(x_{\mathcal{C}})$ are defined on (generally overlapping) subsets of variables $x_{\mathcal{C}} \triangleq \{x_i : i \in \mathcal{C}\}$, and $\mathcal{S} = \{\mathcal{C}\}$.

Inference using the max-product algorithm consists of iteratively passing messages between "connected" random variables of the model; variables of the model that share common factor(s). For a given random variable x_i, the message from variable set $x_{\mathcal{C}\backslash i} \triangleq \{x_j : i \in \mathcal{C}, j \neq i \in \mathcal{C}\}$ to x_i is given by

$$m_{x_{\mathcal{C}\backslash i}\to x_i}(x_i) = \max_{x_{\mathcal{C}\backslash i}} f_{\mathcal{C}}(x_{\mathcal{C}}) \prod_{j\in\mathcal{C}\backslash i} \frac{q(x_j)}{m_{x_{\mathcal{C}\backslash j}\to x_j}(x_j)}, \tag{19}$$

where

$$q(x_i) = \prod_{\mathcal{C}:i\in\mathcal{C}} m_{x_{\mathcal{C}\backslash i}\to x_i}(x_i)$$

is the product of all messages to variable x_i from neighboring variables.

The maximizing arguments of the max operation in (19) are also stored:

$$\hat{x}_{\mathcal{C}\backslash i}(x_i) = \arg\max_{x_{\mathcal{C}\backslash i}} f_{\mathcal{C}}(x_{\mathcal{C}}) \prod_{j\in\mathcal{C}\backslash i} \frac{q(x_j)}{m_{x_{\mathcal{C}\backslash j}\to x_j}(x_j)}. \tag{20}$$

These point from each state of the variable x_i to the maximizing state combination of the variables $x_{\mathcal{C}\backslash i}$.

The product of all the messages coming into any variable x_i provides an estimate of the probability of the maximum a posteriori (MAP) configuration of *all* other variables $x_{j\neq i} = \{x_j : j \neq i\}$ in the probability model, as a function of x_i:

$$q(x_i) \approx k \max_{x_{j\,\neq\,i}} p\left(x_{j\,\neq\,i}, x_i\right). \tag{21}$$

where k is a normalizing constant. Optimization consists of passing messages according to a *message passing schedule*. When the probability model is tree-structured, the global MAP configuration of the variables can be found by propagating messages up and down the tree, and then "decoding", by recursively evaluating $\hat{x}_{\mathcal{C}\backslash i}(x_i)\forall\mathcal{C} : i \in \mathcal{C}$, starting from any x_i. Normally for efficiently, messages are propagated only from the leaves to a chosen root variable. The global MAP configuration can then be obtained by recursively evaluating $\hat{x}_{\mathcal{C}\backslash i}(x_i)$, starting at the root. For HMMs, this procedure reduces to the Viterbi algorithm.

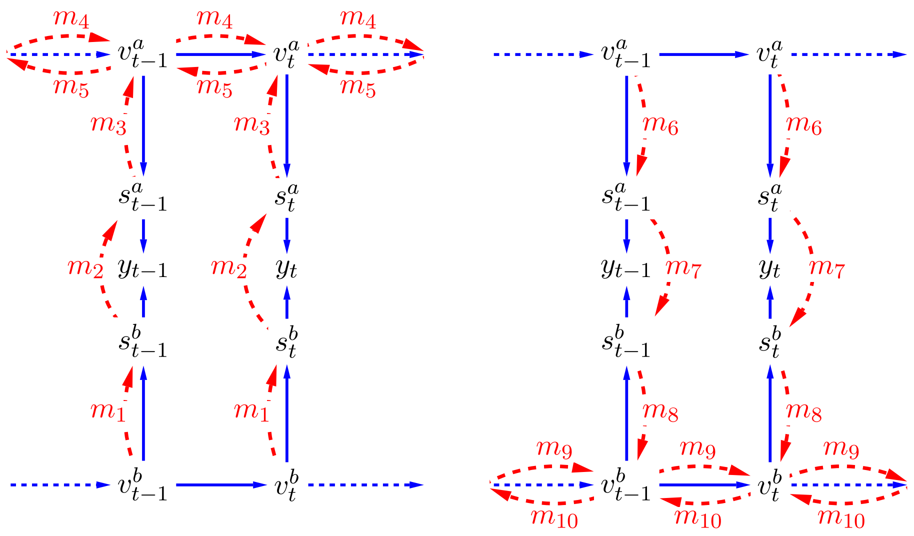
In models such as ours that contain cycles, the messages must be iteratively updated and propagated in all directions. There is, moreover, no guarantee that this approach will converge to the global MAP configuration of the variables. If the algorithm does converge (meaning that all conditional estimates $\hat{x}_{\mathcal{C}\backslash i}(x_i)$ are consistent), the MAP estimate is provably optimal over all tree and single-loop sub-structures of the probability model (Weiss and Freeman, 2001). Convergence, furthermore, is not required to estimate the MAP configuration on any tree sub-structure of the model, which can be obtained in the final iteration by ignoring a given set of dependencies.

Our speech separation model with grammar dynamics (and the speaker features x^a and x^b integrated out), has the form:

$$p\left(\mathbf{y}_{1:T}, s^a_{1:T}, s^b_{1:T}, v^a_{1:T}, v^b_{1:T}\right) = p\left(v^a_1\right)p\left(v^b_1\right)\prod_{t=2}^{T} p\left(v^a_t|v^a_{t-1}\right)p\left(v^b_t|v^b_{t-1}\right)\prod_{t=1}^{T} p\left(s^a_t|v^a_t\right)p\left(s^b_t|v^b_t\right)p\left(\mathbf{y}_t|s^a_t, s^b_t\right), \tag{22}$$

and so the factors $f_{\mathcal{C}}(x_{\mathcal{C}})$ are the conditional probabilities $p(v^a_t|v^a_{t-1})$, $p(s^a_t|v^a_t)$, $p(\mathbf{y}_t|s^a_t, s^b_t)$, and so on.

For the grammar dynamics model, a natural message passing schedule is to alternate between passing messages from one grammar chain to the other, and passing messages along the entire grammar chain of each source (Rennie et al., 2009). This is depicted graphically in Fig. 7. Initially all messages in the graph are initialized to be uniform, and v^a_1 and v^b_1 are initialized to their priors.



(a) Phases 1 and 2 (b) Phases 3 and 4

Fig. 7. Message passing sequences ($m_1 \ldots m_{10}$) on the grammar dynamics model graph. We integrate out x^a and x^b when computing the messages between s^a and s^b. The messages shown in a chain, such as m_4, are passed sequentially along the entire chain, in the direction of the arrows, before moving to the next message. Messages m_6 through m_{10} are the same as m_1 through m_5, but with a and b swapped. Note that m_2 and m_7 are the only messages that involve more than one source model. (a) Phases 1 and 2; (b) phases 3 and 4.

There are four phases of inference:

(1) Pass messages from the grammar states of source b to the grammar states of source a through the interaction function, $p(\mathbf{y}_t|s_t^a, s_t^b)$, for all t (messages m_1 to m_3):

$$m_1(s_t^b) \triangleq m_{v_t^b \to s_t^b} = \max_{v_t^b} p(s_t^b|v_t^b) m_{v_{t-1}^b \to v_t^b} m_{v_{t+1}^b \to v_t^b}$$

$$m_2(s_t^a) \triangleq m_{s_t^b \to s_t^a} = \max_{s_t^b} p(\mathbf{y}_t|s_t^a, s_t^b) m_{v_t^b \to s_t^b}$$

$$m_3(v_t^a) \triangleq m_{s_t^a \to v_t^a} = \max_{s_t^a} p(s_t^a|v_t^a) m_{s_t^b \to s_t^a}$$

(2) Pass messages forward and then backward along the grammar chain for source a, for t=1..T (message m_4), and then backward, for t=T..1 (message m_5):

$$m_4(v_t^a) \triangleq m_{v_{t-1}^a \to v_t^a} = \max_{v_{t-1}^a} p(v_t^a|v_{t-1}^a) m_{v_{t-2}^a \to v_{t-1}^a} m_{s_{t-1}^a \to v_{t-1}^a}$$

$$m_5(v_t^a) \triangleq m_{v_{t+1}^a \to v_t^a} = \max_{v_{t+1}^a} p(v_{t+1}^a|v_t^a) m_{v_{t+2}^a \to v_{t+1}^a} m_{s_{t+1}^a \to v_{t+1}^a}$$

(3) Pass messages from the grammar states of speaker a to the grammar states of speaker b, again via their data interaction (messages m_6 to m_8).

(4) Pass messages forward and backward along the grammar chain for source b, for t=1..T (message m_9), and then backward, for t=T..1 (message m_{10}):

5.2. Max-sum-product algorithm

A variation of the described max-product algorithm is to replace the max operators in the updates for the messages that are sent between the sources with sums (messages m_1, m_2, m_3, m_6, m_7, and m_8). We call the resulting algorithm the *max-sum-product algorithm*. This variation of the algorithm produced substantially better results on the challenge task.

5.3. Dual dynamics

In the dual-dynamics condition we use the full model of Fig. 6. With two speakers, exact inference is computationally complex because the full joint distribution of the grammar and acoustic states, $(v^a \times s^a) \times (v^b \times s^b)$, is required.

Instead we can perform a variety of approximate inference algorithms using belief propagation as described above. One formulation we tried involved alternating the 2-D Viterbi search between two factors: the Cartesian product $s^a \times s^b$ of the acoustic state sequences and the Cartesian product $v^a \times v^b$ of the grammar state sequences. Again, in the same-talker condition, the 2-D Viterbi search breaks the symmetry in each factor. A variation of this involves iterating 2-D forward–backward iterations on the two chains, followed by Viterbi steps. In addition, we experimented with the max-product algorithm on the graph of Fig. 4. None of these variations outperformed the 2-D Viterbi algorithm on the grammar dynamics model. However we did not exhaustively search the space of these models and we still see this as a useful direction to explore.

6. Acoustic likelihood estimation

Model-based speech separation would be impractical without special techniques to reduce computation time. The exact 2-D Viterbi inference requires a large number of state combinations to be evaluated. To speed up the evaluation of the joint state likelihood, in addition to sharing the same acoustic Gaussians across all grammar states, we also employed both *band quantization* of the acoustic Gaussians and *joint-state pruning*. These optimizations are needed because the interaction function generally requires all combinations of acoustic states to be considered. The max interaction model is inherently faster than Algonquin and can also be approximated to further reduce computational cost.

6.1. Band quantization

One source of computational savings stems from the fact that some of the Gaussians in our model may differ only in a few features. Band quantization addresses this by approximating each of the D_s Gaussians of each model with a shared set of d Gaussians, where $d \ll D$, in each of the F frequency bands of the feature vector. A similar idea is described in Bocchieri (1993). For a diagonal covariance matrix, $p(\mathbf{x}^a|s^a) = \prod_f N(x^a_f; \mu_{f,s^a}, \sigma^2_{f,s^a})$, where σ^2_{f,s^a} are the diagonal elements of covariance matrix $\mathbf{\Sigma}_{s^a}$. The mapping $M_f(s^i)$ associates each of the D_s Gaussians with one of the d Gaussians in band f. Now $\hat{p}(\mathbf{x}^a|s^a) = \prod_f N(x^a_f; \mu_{f,M_f(s^a)}, \sigma^2_{f,M_f(s^a)})$ is used as a surrogate for $p(\mathbf{x}^a|s^a)$. Fig. 8 illustrates the idea.

Under this model the d Gaussians are optimized by minimizing the KL-divergence $D_s(\sum_{s^a} p(s^a)p(\mathbf{x}^a|s^a)||\sum_{s^a} p(s^a)\hat{p}(\mathbf{x}^a|s^a))$, and likewise for s^b, using the variational approximation of Hershey and Olsen (2007). Then in each frequency band, only $d \times d$, instead of $D_s \times D_s$ combinations of Gaussians have to be evaluated to compute $p(\mathbf{y}|s^a, s^b)$. Despite the relatively small number of components d in each band, taken across bands, band quantization is capable of expressing d^F distinct patterns, in an F-dimensional feature space. In practice only a subset of these will be used to approximate the Gaussians in a given model. We used $d = 8$ and $D_s = 256$, which reduced the likelihood computation time by several of magnitude.

6.2. Joint state pruning

Another source of computational savings comes from the sparseness of the model. Only a handful of s^a, s^b combinations have likelihoods that are significantly larger than the rest for a given observation. Only these states are required to adequately explain the observation. By pruning the total number of combinations down to a smaller number we can speed up the temporal inference.

We must estimate all likelihoods in order to determine which states to retain. We therefore used band quantization to estimate likelihoods for all states, perform state pruning, and then evaluate the full likelihood model on the pruned states using the exact parameters. In the experiments reported here, we pruned down to 256 state combinations.

Please cite this article in press as: Hershey, J.R. et al., Super-human multi-talker speech recognition: ..., Computer Speech and Language (2009), doi:10.1016/j.csl.2008.11.001

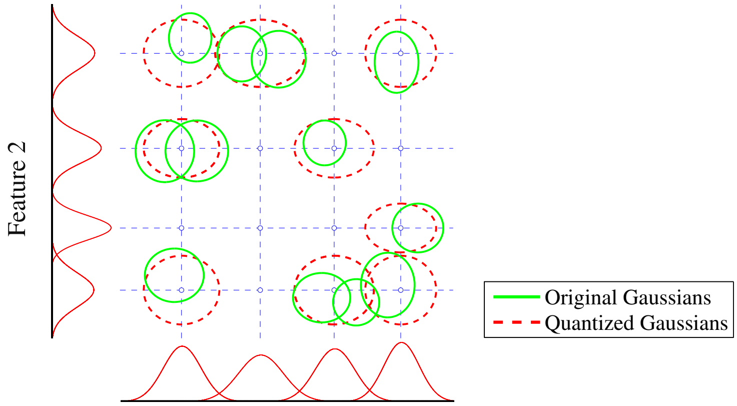


Fig. 8. In band quantization, a large set of multi-dimensional Gaussians is represented using a small set of shared unidimensional Gaussians optimized to best fit the original set of Gaussians. Here we illustrate twelve two-dimensional Gaussians (green ellipses). In each dimension we quantize these to a pool of four shared unidimensional Gaussians (red density plots on axes). The means of these are drawn as a grid (blue dashed lines), on which the quantized two-dimensional Gaussians (red dashed ellipses) can occur only at the intersections. Each quantized two-dimensional Gaussian is constructed from the corresponding pair of unidimensional Gaussians, one for each feature dimension. In this example we represent 24 means and variances (12 Gaussians ×2 dimensions), using 8 means and variances (4 Gaussians ×2 dimensions). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

6.3. Marginal likelihoods

The max-sum-product loopy belief propagation algorithm presented in this paper requires that the following messages be iteratively computed:

$$m_2(s_t^a) = \sum_{s_t^b} p(\mathbf{y}_t|s_t^a, s_t^b) m_1(s_t^b), \quad \text{and} \quad m_7(s_t^b) = \sum_{s_t^a} p(\mathbf{y}_t|s_t^a, s_t^b) m_6(s_t^a),$$

These updates couple inference over the speakers, and require $O(D_s^2)$ operations per message, because all acoustic state combinations must be considered. This is the case for both Algonquin and the max model. Under the max model, however, the data likelihood in a single frequency band (10) consists of N terms, each of which *factor* over the acoustic states of the sources. Currently we are investigating linear-time algorithms ($O(ND_s)$) that exploit this property to approximate $p(y_k^s)$.

6.4. Max model approximation

For many pairs of states one model is significantly louder than another $\mu_{s_t^a} \gg \mu_{s_t^b}$ in a given frequency band, relative to their variances. In such cases we can closely approximate the max model likelihood as $p(y_t|s_t^a, s_t^b) \approx p_{x_t^a}(y_t|s_t^a)$, and the posterior expected values according to $E(x_t^a|y_t, s_t^a, s_t^b) \approx y_t$ and $E(x_t^b|y_t, s_t^a, s_t^b) \approx \min(y_t, \mu_{s_t^b})$, and similarly for $\mu_{s_t^a} \ll \mu_{s_t^b}$. In our experiments this approximation made no significant difference, and resulted in significantly faster code by avoiding the Gaussian cumulative distribution function. It is therefore used throughout this paper in place of the exact max algorithm.

7. Speaker and gain estimation

The identities of the two speakers comprising each multi-talker test utterance are unknown at test time, and therefore must be estimated when utilizing speaker-dependent acoustic models.

The SNR of the target speaker relative to the masker varies from 6 dB to −9 dB in the test utterances. We trained our separation models on gain-normalized speech features, so that more representative acoustic models could be learned using diagonal-covariance GMMs, and so the absolute gains of each speaker also need to be inferred.

Please cite this article in press as: Hershey, J.R. et al., Super-human multi-talker speech recognition: ..., Computer Speech and Language (2009), doi:10.1016/j.csl.2008.11.001

The number of speakers and range of SNRs in the multi-talker test set makes it expensive to directly consider every possible combination of models and gains. If the speaker gains are each quantized to 3 bits, and we use 34, 256-component speaker-dependent acoustic models to represent the speakers, for example, there are $(34 \cdot 8)^2 > 2^{16}$ possible speaker/gain configurations to search over, each with $256^2 = 2^{16}$ acoustic states that need to be evaluated at each time step.

We avoid this computation by doing a model-based analysis of each test utterance, that assumes that only one speaker emits each observed signal frame. Under the model, frames that are dominated by a single talker, and have distinguishing features, will have sharp speaker posteriors. These frames are identified and used to narrow down what speakers are present in the mixture. We then explore the greatly reduced set of plausible speaker combinations with an approximate EM procedure, to select a single pair of speakers, and optimize their gains.

We model the observed features at frame t as generated by a single speaker c, and assume that log spectrum of the speaker is described by a mixture model:

$$p(\mathbf{y}_t, c) = p(c) \sum_g p(g) \sum_{s^c} p(s^c|c) p(\mathbf{y}_t|s^c, g) \tag{23}$$

where the speaker gain g is modeled as discrete and assumed to be uniformly distributed over $\mathcal{G} = \{6, 3, 0, -3, -6, -9, -12\}$, and $p(\mathbf{y}_t|s^c, g) = \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_{s^c} + g, \boldsymbol{\Sigma}_{s^c})$, where $\boldsymbol{\mu}_{s^c}$ and $\boldsymbol{\Sigma}_{s^c}$ are the gain-normalized mean and variance of component s^c of speaker c. The speaker prior $p(c) = \pi_c$ is initialized to be uniform.

To estimate the posterior distribution of each source in the mixture, we apply the following simple EM algorithm. In the E-Step, we compute the posterior distribution of each source for each frame, given the current parameters of the model:

$$p(c|\mathbf{y}_t) = \frac{p(\mathbf{y}_t, c)}{\sum_c p(\mathbf{y}_t, c)} \tag{24}$$

In the M-Step, we update the parameters of the source prior, $\{\pi_c\}_c$, given $\{p(c|\mathbf{y}_t)\}_t$. To make the procedure robust in multi-talker data, only frames $t \epsilon \mathcal{T}$, where the uncertainty in the generating speaker is low are considered in the parameter update:

$$\pi_c = E_{\mathcal{T}}[p(c|\mathbf{y}_t)] = \frac{1}{|\mathcal{T}|} \sum_{t \epsilon \mathcal{T}} p(c|\mathbf{y}_t), \tag{25}$$

where $\mathcal{T} = \{t : \max_c p(c|\mathbf{y}_t) > \alpha\}$, and α is a chosen threshold. In this manner frames with features that are common to multiple sources (such as silence), and frames that are comprised of multiple sources, and therefore not well explained by any single speech feature, are not included in the speaker parameter updates.

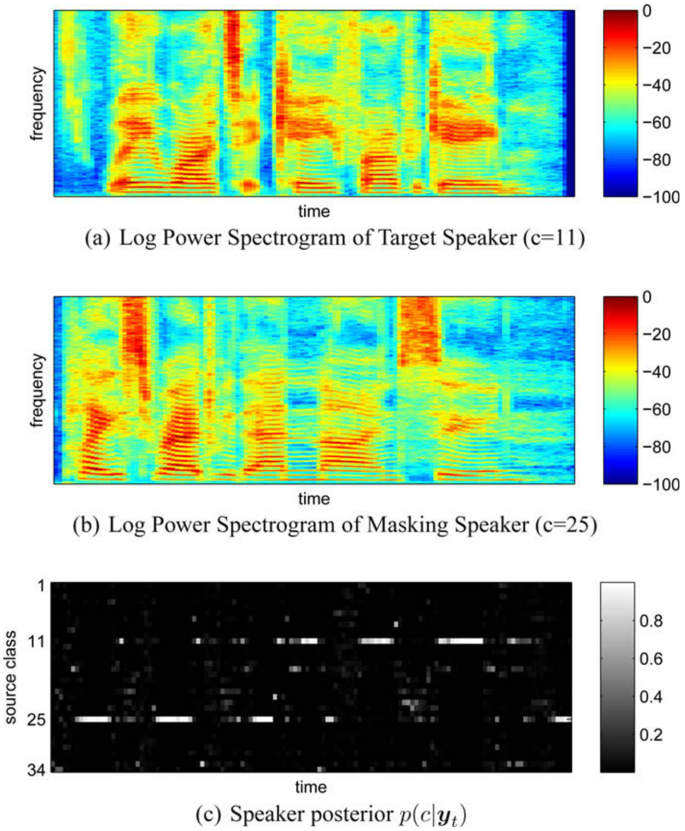
The updates may be iterated until convergence, but we have found that a single EM iteration suffices. The posterior distribution of the speakers given the entire test utterance is taken as $\{\pi_c\}$, which is the expected value of the posterior distribution of the speakers, taken over frames that are dominated by a single source and have distinguishing features.

Fig. 9 depicts the original spectrograms of the target and masker signals and the speaker posteriors $p(c|\mathbf{y}_t)$ plotted as a function of t, for a typical test mixture in the challenge two-talker corpus. The speaker posteriors are sharply peaked in regions of the mixture where one source dominates.

Given a set of speaker finalists chosen according to $\{\pi_c\}$, we apply the following approximate EM algorithm, to each speaker combination $\{a, b\}$, to identify what speakers are present in the mixture and adapt their gains. We use the approximate max model (see Section 6.4) to compute likelihoods for this algorithm, although similar can be derived for the other interaction models. The speaker combination whose gain-adapted model combination maximizes the probability of the test utterance is selected for utilization by the separation system.

(1) E-Step: Compute the state posteriors $p^i(s_t^a, s_t^b|\mathbf{y}_t)$ for all t given the current speaker gain estimates g_a^{i-1} and g_b^{i-1}, where i is the iteration index, using the max approximation (see Section 4),

(2) M-Step: Update the gain estimates given the computed state posteriors. The update for g_a^i is



(a) Log Power Spectrogram of Target Speaker (c=11)

(b) Log Power Spectrogram of Masking Speaker (c=25)

(c) Speaker posterior $p(c|\boldsymbol{y}_t)$

Fig. 9. Plots of the (unobserved) spectrograms of the target and masker speakers and the speaker posteriors $p(c|\mathbf{y}_t)$ under the single source emission model, for a typical test utterance in the two-talker corpus (mixed at 0 dB).

$$g_a^i = g_a^{i-1} + \alpha_i \Delta g_a^i \tag{26}$$

$$\Delta g_a^i = \frac{\sum_t \sum_{s_t^a,s_t^b} p^i(s_t^a,s_t^b|\mathbf{y}_t) \sum_{f\in\mathcal{F}_{s_t^a>s_t^b}} \frac{\Delta g^i_{f,s_t^a}}{\sigma^2_{f,s_t^a}}}{\sum_t \sum_{s_t^a,s_t^b} p^i(s_t^a,s_t^b|\mathbf{y}_t) \sum_{f\in\mathcal{F}_{s_t^a>s_t^b}} \frac{1}{\sigma^2_{f,s_t^a}}} \tag{27}$$

where $\Delta g^i_{f,s_t^a} = (y_{f,t} - \mu_{f,s_t^a} - g_a^{i-1})$, $\mathcal{F}_{s_t^a>s_t^b}$ is the set of frequency bins where $\mu_{s_t^a} + g_a^{i-1} > \mu_{s_t^b} + g_b^{i-1}$ (where the gain-adapted feature of source a is greater than that of source b), and α_i is the adaptation rate.Here μ_{f,s_t^a} and σ^2_{f,s_t^a} represent the mean and variance of component s of speaker model a at frequency bin f, respectively. The g_b^i update is analogous.

Note that the probability of the data is not guaranteed to increase at each iteration of this EM procedure, even when $\alpha_i = 1$, because in the approximate max model, the joint state posterior $p^i(s_t^a, s_t^b|\mathbf{y}_t)$ is not continuous in g_a^i and g_b^i, and so the dimension assignment $\mathcal{F}_{s_t^a>s_t^b}$ changes depending on the current gain estimate. Empirically however, this approach has proved to be effective.

Table 2 reports the speaker identification accuracy obtained on the two-talker test set via this approach, when all combinations of the most probable source and the six most probable sources are considered (six combinations total), and the speaker combination maximizing the probability of the data is selected. Over all mixture cases and conditions on the two-talker test set we obtained greater than 98% speaker identification accuracy overall.

Table 2
Speaker identification accuracy (percent) as a function of test condition and case on the two-talker test set, for the presented source identification and gain estimation algorithm.

Condition	6 dB	3 dB	0 dB	−3 dB	−6 dB	−9 dB	All
Same talker	100	100	100	100	100	99	99.8
Same gender	97	98	98	97	97	96	97.1
Different gender	99	99	98	98	97	96	97.6
All	98.4	99.1	99.0	98.2	98.1	96.5	98.2

Table 3
Word error rates (percent) on the stationary noise development set. The error rate for the “random-guess” system is 87%. The systems in the table are: (1) The default HTK recognizer, (2) IBM–GDL MAP–adapted to the speech separation training data, (3) MAP–adapted to the speech separation training data and artificially generated training data with added noise, (4) Oracle MAP adapted speaker-dependent system with known speaker IDs at test time, (5) MAP adapted speaker-dependent models with SDL, and (6) human listeners.

System	Noise condition				
	Clean	6 dB	0 dB	−6 dB	−12 dB
HTK	***1.0***	45.7	82.0	88.6	87.2
GDL–MAP I	2.0	33.2	68.6	85.4	87.3
GDL–MAP II	2.7	7.6	14.8	49.6	77.2
Oracle	1.1	4.2	8.4	39.1	***76.4***
SDL	1.4	***3.4***	***7.7***	***38.4***	77.3
Human	0.6	1.7	5.0	30.7	62.5

8. Recognition using speaker-dependent labeling (SDL)

When separating the mixed speech of two speakers, we start with an estimated speaker ID and gain combination, say speaker *a* and speaker *b*. However, we do not yet know which speaker is saying white. So, in the models with grammars we separate under two hypotheses: H1, in which speaker *a* says “white” and speaker *b* does not, and H2, in which speaker *b* says “white” and speaker *a* does not. We use a grammar containing only “white” for speakers hypothesized to say “white”, and one that contains the other three colors for speakers hypothesized not to say “white”. Likewise, we perform SDL recognition on each pair of outputs under the same hypothesis that generated it, using the same grammar for each sequence. The system then picks the hypothesis that yielded the highest combined likelihood for the target and masker pair.

Our speech recognition system uses speaker-dependent labeling (SDL) (Rennie et al., 2006) to do rapid speaker adaptation. This method uses speaker-dependent models for each of the 34 speakers. Instead of using the speaker identities provided by the speaker ID and gain module, we followed the approach for gender dependent labeling (GDL) described in Olsen and Dharanipragada (2003). This technique provides better results than if the true speaker ID is specified.

Incidentally, the grammar-based version of the separation system also provides a hypothesis of the utterance text. Nevertheless, the SDL recognizer still produced better results from the reconstructed signals. This may be because the recognizer uses better features for recognizing clean speech. Another explanation might be that the separation system may make mistakes in estimating the words, but as long as it correctly estimates the times and frequencies where one signal dominates the other, the original signal will be reconstructed correctly.

We employed MAP training Gauvain and Lee, 1994 to train a speaker-dependent model for each of the 34 speakers. The speech separation challenge also contains a stationary colored noise condition, which we used to test the noise-robustness of our recognition system. The performance obtained using MAP adapted speaker-dependent models with the baseline gender dependent labeling system (GDL) and SDL are shown in Table 3. The SDL technique (described below) achieves better results than the MAP adapted system using oracle knowledge of the speaker ID.

8.1. Theory of SDL

Instead of using the speaker identities provided by the speaker ID and gain module directly in the recognizer, we followed the approach for gender dependent labeling (GDL) described in Olsen and Dharanipragada (2003).

Each speaker c is associated with a set, $\mathcal{S}_c$, of 39 dimensional cepstrum domain acoustic Gaussian mixture models. At a particular time frame then we have the following estimate of the *a posteriori* speaker probability given the speech feature $\mathbf{x}_t$:

$$p(c|\mathbf{x}_t) = \frac{\sum_{s\in\mathcal{S}_c}\pi_s\mathcal{N}(\mathbf{x}_t;\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s)}{\sum_{c'}\sum_{s\in\mathcal{S}_{c'}}\pi_s\mathcal{N}(\mathbf{x}_t;\boldsymbol{\mu}_s,\boldsymbol{\Sigma}_s)}.$$

SDL does not make the assumption that each file contains only one speaker, but instead assumes only that the speaker identity is constant for a short time, and that the observations are unreliable. The speaker probability is thus averaged over a time window using the following recursive formula:

$$p(c|\mathbf{x}_{1:t}) \overset{\text{def}}{=} \alpha p(c|\mathbf{x}_{1:t-1}) + (1-\alpha)p(c|\mathbf{x}_t) \tag{28}$$

for speaker c at time t, and where α is a time constant. This is equivalent to smoothing the frame-based speaker posteriors using the following exponentially decaying time window.

$$p(c|\mathbf{x}_{1:t}) = \sum_{t\prime=1}^{t}(1-\alpha)\alpha^{t-t'}p(c|\mathbf{x}_{t'}), \tag{29}$$

The effective window size for the speaker probabilities is given by $\alpha/(1-\alpha)$, and can be set to match the typical duration of each speaker. We chose $\alpha/(1-\alpha) = 100$, corresponding to a speaker duration of 1.5 s.

The online *a posteriori* speaker probabilities are close to uniform even when the correct speaker is the one with the highest probability. We can remedy this problem by sharpening the probabilities. The boosted speaker detection probabilities are defined as

$$\pi_c(t) = p(c|\mathbf{x}_{1:t})^{\beta}\Big/\sum_{c'}p(c'|\mathbf{x}_{1:t})^{\beta}. \tag{30}$$

We used $\beta = 6$ for our experiments. During recognition we can now use the boosted speaker detection probabilities to give a time-dependent Gaussian mixture distribution:

$$\text{GMM}(\mathbf{x}_t) = \sum_c \pi_c(t)\text{GMM}_c(\mathbf{x}_t).$$

As can be seen in Table 3, despite having to infer the speaker IDs, the SDL system outperforms the oracle system, which knows the speaker ID at test time.

9. Experimental results

Human listener performance is compared in Fig. 10 to results using the SDL recognizer without speech separation, and for each of the proposed models. Performance is poor in all conditions when separation is not used. With separation, but no dynamics, the models do surprisingly well in the different talker conditions, but poorly when the signals come from the same talker. Acoustic dynamics give some improvement, mainly in the same talker condition. The grammar dynamics model seems to give the most benefit, bringing the overall error rate below that of humans. The dual-dynamics model performed about the same as the grammar dynamics model. However, it remains to be seen if tuning the relative weight of the grammar versus acoustic dynamics improves results in the dual-dynamics model.

Fig. 11 shows the relative word error rate of the best system compared to human subjects. For SNRs in the range of 3 dB to −6 dB, the system exceeds human performance. In the same-gender condition when the speakers are within 3 dB of each other the system makes less than half the errors of the humans. Human listeners do better when the two signals are at different levels, even if the target is below the masker

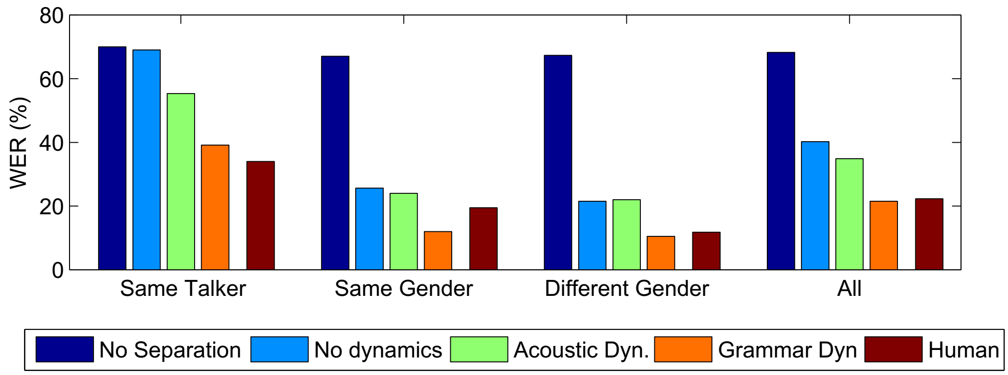


Fig. 10. Average word error rate (WER) as a function of model dynamics, in different talker conditions, compared to Human error rates, using Algonquin.

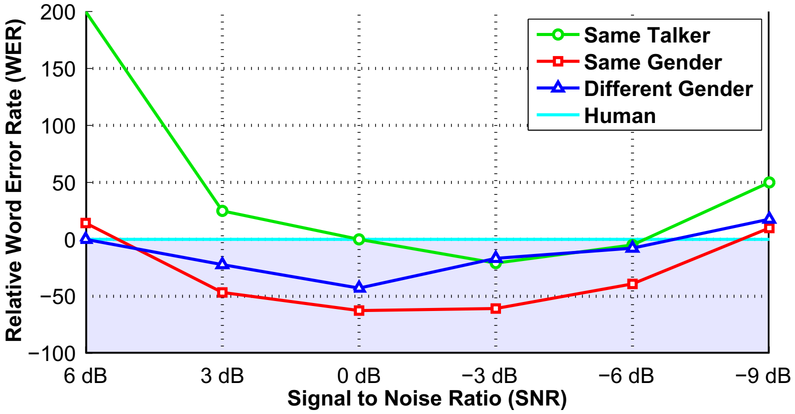


Fig. 11. Word error rate of best system relative to human performance. That is, a relative WER of −50% corresponds to half as many errors as humans, a relative WER of 0% is human performance, and a relative WER of 100% indicates twice as many errors as humans. Shaded area is where the system outperforms human listeners.

Table 4
Error rates for the Algonquin algorithm with grammar dynamics. Cases where the system performs as well as, or better than, humans are emphasized.

Condition	6 dB	3 dB	0 dB	−3 dB	−6 dB	−9 dB	Total
Same talker	30	35	***46***	***38***	***38***	48	39.3
Same gender	8	***8***	***9***	***11***	***14***	22	***11.9***
Different gender	***6***	***7***	***8***	***10***	***12***	20	***10.7***
Overall	15.4	***17.8***	***22.7***	***20.8***	***22.1***	30.9	***21.6***

(i.e., in −9 dB), suggesting that they are better able to make use of differences in amplitude as a cue for separation.

Table 4 shows the results of one of the best overall systems, which uses the Algonquin model and grammar dynamics. In the above experiments, the clean condition was not considered in the overall result, as it was not a requirement in the challenge.

Table 5
Comparison of system with *clean detection* against baseline without clean detection for clean conditions and two-talker conditions (averaged across all SNRs).

System	Condition	
	Clean	Two-talker
Baseline	26.6	21.6
Clean detection	9.6	21.6

9.1. Handling clean conditions

In clean conditions, the original system performs poorly, at an error rate of 26.6%, because the gain estimation system assumes there are two speakers. To show that we can handle the clean situation even with poor gain estimates, we introduce a maximum-likelihood *clean detection* method, as follows: In addition to the hypotheses, H1 and H2, about which estimated speaker is the target, we add a third hypothesis, H3, that there is only one speaker. Under this hypothesis we recognize the mixed speech directly using the SDL recognizer, measuring its likelihood. We then compared this likelihood with the averaged likelihoods of the best pair of enhanced signals. The final decoded result is then the one with the best overall likelihood. Results using clean detection in Table 5 show that it improves performance in clean conditions without hurting performance in two-talker conditions.

9.2. Speaker independent background models

In the experiments reported above we take full advantage of the closed speaker set. In most scenarios it is unrealistic to assume any knowledge of the set of background speakers. However, there are some scenarios where the target speaker may be entirely known. This motivates an experiment in which we use a speaker-dependent target model, and a speaker independent background model. Here we consider two cases of background model: one that is entirely speaker independent, and another that is gender dependent. To remove the influence of the speaker ID and gain estimation algorithm, here we compare results using the oracle speaker IDs and gains for the target and masker. In both cases all the background models 256 acoustic states each. Table 6 gives the results. As expected, decreasing the specificity of the background model increases error rates in every condition. However, the more general background models are using fewer states relative to the number of speakers being represented, so this may also be a significant factor in the degradation.

9.3. Background grammar

Another interesting question is how important the grammar constraints are. Above, we tested systems with different levels of constraints on dynamics, and those without a grammar fared poorly for this highly constrained task, relative to those with the task grammar. In realistic speech recognition scenarios, however, generally little is known about the background speaker's grammar. To relax the grammar constraints we used a “bag of words” for the masker grammar, which consisted of an unordered collection of *all* words in the grammar (including “white”). Using the grammar-dynamics model and oracle speaker IDs and gains, the overall error rate was 23.2%, compared with 19.0% for the grammar used in the main experiment. It would be interesting to further relax the masker dynamics to a phoneme bi-gram model or even just acoustic-level dynamics, for instance.

9.4. Known transcripts

At the opposite extreme, an interesting question is to what extent tighter grammar constraints might affect the results. In some scenarios the actual transcript is known. For example, this is the case with a closed-captioned movie soundtrack, a song with background music, or a transcribed meeting. Even though the transcription is known it might be useful to extract the original voices from the mixture in an intelligible way for human

Table 6
Comparison of results of the Algonquin method, for known target speaker, with grammar dynamics for three background model types: speaker-dependent (SD) with known masker, gender dependent (GD) with known masker gender, and speaker independent (SI). Oracle gains were used in all cases.

Condition	SD	GD	SI
Same talker	33.3	41.5	57.5
Same gender	11.5	12.8	21.6
Different gender	9.9	11.9	15.6
Overall	19.0	23.1	32.8

Table 7
WER as a function of separation algorithm and test condition. In all cases oracle speaker IDs and gains were used, and Algonquin was used to approximate the acoustic likelihoods unless otherwise noted. The *joint Viterbi* algorithm scales exponentially with the number of sources. The iterative loopy belief propagation algorithms, on the other hand, scale *linearly* with language model. Results exceeding human performance are emphasized.

Inference		Joint	Max	Iterative	Iterative	
Algorithm	Human	Viterbi	Product	Viterbi	Max-sum product	
Likelihoods	?	Algonquin	Algonquin	Algonquin	Algonquin	Max
ST	34.0	***33.3***	42.0	44.3	39.7	38.6
SG	19.5	***11.5***	***12.9***	***16.4***	***12.0***	***14.4***
DG	11.9	***9.9***	12.0	13.9	***11.1***	***10.8***
Overall	22.3	***19.0***	23.3	25.8	***21.9***	***22.1***

listening. To test this scenario, we limited the grammars in the separation model to just the reference transcripts for each speaker. To measure intelligibility, we measured recognition on the separated target signal using the SDL recognizer with the full task grammar. The error rate on the estimated sources dropped from 19.0% down to 7.6% overall. This is consistent with an improvement to human intelligibility, and informal listening tests confirmed that the source signals were typically extracted remarkably well, and in a way that preserved the identity and prosody of the original speaker.

9.5. Iterative source estimation

To see how well we can do with a separation algorithm that scales well to larger problems – with more speakers, a larger vocabulary, and so on – we experimented with the use of loopy belief propagation to avoid searching over the joint grammar state–space of the speaker models.

Table 7 summarizes the WER performance of the system as a function of separation algorithm. For all iterative algorithms, the message passing schedule described in Section 5 was executed for 10 iterations to estimate the most likely configuration of the grammar states of both sources. After inferring the grammar state sequences, MMSE estimates of the sources were then reconstructed by averaging over all active acoustic states. In all cases, oracle speaker IDs and gains were used.

The *iterative Viterbi* algorithm is equivalent to the iterative max-sum product algorithm, but with the messages from the grammar to the acoustic states of each source bottlenecked to the single maximum value.

The proposed iterative message-passing algorithms perform comparably to the *joint Viterbi* algorithm, which does exact temporal inference. Interestingly, the results obtained using the iterative max-sum product algorithm are significantly better than those of the max-product algorithm, presumably because this leads to more accurate grammar state likelihoods.

Temporal inference using the max-sum product algorithm is significantly faster than the exact temporal inference, and still exceeds the average performance of human listeners on the task. These iterative algorithms, moreover, scale linearly with language model size. Table 8 summarizes the WER performance and relative number of operations required to execute each algorithm as a function of grammar beam size and acoustic

Table 8
WER and relative number of operations as a function of algorithm, likelihood model, and beam size.

Inference Algorithm	Joint Viterbi	Joint Viterbi	Iterative Max-sum product	
Likelihoods	Algonquin	Algonquin	Algonquin	Max
Beam size	20000	400	Full	Full
Task error rate	19.0	22.1	21.9	22.1
Temporal inference (relative operations)	10X	3X	1X	1X

interaction model. Here *temporal inference* refers to all computation explicitly associated with the source grammars (including the acoustic likelihood to grammar mapping). Even for two sources, temporal inference with loopy belief propagation is three times more efficient than joint Viterbi with a beam of 400, which yields comparable WER performance.

10. Conclusion

We have described a system for separating and recognizing speech that outperforms human listeners on the monaural speech separation and recognition challenge. The computation required for exact inference in the best system is exponentially complex in the number of sources and the size of the models. However, we have shown that using approximate inference techniques, we can perform nearly the same temporal inference with complexity that is linear in number of sources and the size of the language models. The approach can therefore be readily scaled to more complex problems. Of course, many problems must be solved in order to make model-based speech separation viable for real-world applications. There is room for improvement in the models themselves, including separation of excitation and filter dynamics, adaptation to unknown speakers and environments, better modeling of signal covariances, and incorporating phase constraints. An other important extension is to use microphone arrays to help improve the separation wherever possible. Perhaps the most important direction of future research is to further reduce the computational cost of inference, especially in the evaluation of the acoustic likelihoods. We are currently investigating algorithms for computing the marginal acoustic likelihoods, which, in combination with the loopy belief propagation methods introduced here, would make the complexity of the entire system linear in the number of speakers and states.

References

Bocchieri, E., 1993. Vector quantization for the efficient computation of continuous density likelihoods. In: ICASSP, vol. II. pp. 692–695.

Cooke, M., Barker, J., Cunningham, S., Shao, X., 2006. An audio-visual corpus for speech perception and automatic speech recognition. Journal of the Acoustical Society of America 120, 2421–2424.

Cooke, M., Hershey, J.R., Rennie, S.J., 2009. The speech separation and recognition challenge. Computer Speech and Language, this issue.

Ephraim, Y., 1992. A Bayesian estimation approach for speech enhancement using hidden Markov models. IEEE Transactions on Signal Processing 40 (4), 725–735.

Frey, B.J., Deng, L., Acero, A., Kristjansson, T., 2001. Algonquin: Iterating laplace's method to remove multiple types of acoustic distortion for robust speech recognition, proceedings of Eurospeech.

Gales, M., Young, S., 1996. Robust continuous speech recognition using parallel model combination. IEEE Transactions on Speech and Audio Processing 4 (5), 352–359.

Gauvain, J., Lee, C., 1994. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Transactions on Speech and Audio Processing 2 (2), 291–298.

Ghahramani, Z., Jordan, M.I., 1995. Factorial hidden Markov models. In: Advances in Neural Information Processing Systems, vol. 8.

Hershey, J.R., Olsen, P.A., 2007. Approximating the Kullback Leibler divergence between gaussian mixture models. In: ICASSP. Honolulu, Hawaii.

Hershey, J.R., Kristjansson, T.T., Rennie, S.J., Olsen, P.A., 2006. Single channel speech separation using factorial dynamics. In: Advances in Neural Information Processing Systems, vol. 19, December 4–7, Vancouver, British Columbia, Canada.

Kristjansson, T.T., Attias, H., Hershey, J.R., 2004. Single microphone source separation using high-resolution signal reconstruction. In: ICASSP.

Kschischang, F., Frey, B., Loeliger, H., 2001. IEEE Transactions on Information Theory 47 (2), 498–519.

Linde, Y., Buzo, A., Gray, R.M., 1980. An algorithm for vector quantizer design. IEEE Transactions on Communications 28 (1), 84–95.

Please cite this article in press as: Hershey, J.R. et al., Super-human multi-talker speech recognition: ..., Computer Speech and Language (2009), doi:10.1016/j.csl.2008.11.001

Nádas, A., Nahamoo, D., Picheny, M.A., 1989. Speech recognition using noise-adaptive prototypes. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 37. pp. 1495–1503.

Olsen, P.A., Dharanipragada, S., 2003. An efficient integrated gender detection scheme and time mediated averaging of gender dependent acoustic models proceedings of Eurospeech. vol. 4. pp. 2509–2512.

Radfar, M.H., Dansereau, R.M., Sayadiyan, A., 2006. Nonlinear minimum mean square error estimator for mixture–maximisation approximation. Electronics Letters 42 (12), 724–725.

Rennie, S.J., Olsen, P.A., Hershey, J.R., Kristjansson, T.T., 2006. Separating multiple speakers using temporal constraints. In: ISCA Workshop on Statistical And Perceptual Audition.

Rennie, S.J., Hershey, J.R., Olsen, P.A., 2009. Single-channel speech separation and recognition using loopy belief propagation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing.

Roweis, S., 2003. Factorial models and refiltering for speech separation and denoising. Eurospeech, 1009–1012.

Varga, P., Moore, R.K., 1990. Hidden Markov model decomposition of speech and noise. In: ICASSP. pp. 845–848.

Virtanen, T., 2006. Speech recognition using factorial hidden Markov models for separation in the feature space. In: ICSLP.

Weiss, Y., Freeman, W.T., 2001. On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs. IEEE Transactions on Information Theory 47 (2), 736–744.

Please cite this article in press as: Hershey, J.R. et al., Super-human multi-talker speech recognition: ..., Computer Speech and Language (2009), doi:10.1016/j.csl.2008.11.001