Model-Based Multiple Pitch Tracking Using Factorial HMMs: Model Adaptation and Inference

Michael Wohlmayr, *Member, IEEE,* and Franz Pernkopf, *Member, IEEE*

***Abstract*—Robustness against noise and interfering audio signals is one of the challenges in speech recognition and audio analysis technology. One avenue to approach this challenge is single-channel multiple-source modeling. Factorial hidden Markov models (FHMMs) are capable of modeling acoustic scenes with multiple sources interacting over time. While these models reach good performance on specific tasks, there are still serious limitations restricting the applicability in many domains.**

In this paper, we generalize these models and enhance their applicability. In particular, we develop an EM-like iterative adaptation framework which is capable to adapt the model parameters to the specific situation (e.g. actual speakers, gain, acoustic channel, etc.) using only speech mixture data. Currently, source-specific data is required to learn the model. Inference in FHMMs is an essential ingredient for adaptation. We develop efficient approaches based on observation likelihood pruning. Both adaptation and efficient inference are empirically evaluated for the task of multipitch tracking using the GRID corpus.

***Index Terms*—Multipitch tracking, factorial hidden Markov model, mixture maximization, Gaussian mixture model, efficient inference, model adaptation, self-adaptation**

I. Introduction

Speech recognition is performing reliably in acoustically clean environments. However, in harsh environments where the speech signal is distorted by interfering acoustic sources, speech recognition performs far from satisfactory. These difficult situations are well-known as *cocktail party problem* [1]. The human auditory system has a remarkable ability to focus on a single talker, even in the presence of interfering acoustic sources. This can be partially attributed to the binaural processing ability. However, even in the monaural, i.e. single-channel, case, where the auditory system can not rely on direction specific information, scenarios with multiple interfering sources are deciphered remarkably well by humans.

Multi-channel approaches are commonly used to enhance the target signal or to perform source separation [2]. Multiple microphones enable to exploit spatial features of the sound field for speech enhancement in addition to the temporal information, i.e. signals are attenuated according to their location in space. However, in diffuse noise fields, spatial filtering alone does not result in sufficient noise suppression. Therefore, the beam-former output is usually enhanced in a post-processing stage. Classical *single-channel* speech enhancement [3] is applied for denoising, i.e. the interfering audio source is suppressed. In contrast, the emphasis of this paper is on modeling *multiple* single-channel audio sources, i.e. the focus is on the acoustic scene. For cases, e.g. *media mining*, where the available data has *not* been recorded with multiple microphones, single-channel multiple-source modeling techniques are important and can be also used as post-processing in the multi-microphone setting. Potential applications are preprocessing for automatic speech recognition in noisy and multitalker environments, word spotting, media-mining, or audio information retrieval. Extracting and modeling sources from a single-channel mixture is in general ill-posed. Most methods belong to one of the following categories: *(i) Computational auditory scene analysis (CASA):* CASA [4] aims to mimic low-level separation and grouping mechanisms of the biological auditory system. It works bottom-up, where spectro-temporal components that are likely to belong to the same source are combined to form output streams of individual sources. *(ii) Model-based approaches:* Typically, model-based approaches are trained a-priori on a time-frequency representation of *clean source specific* data. During separation the model is applied in a top-down fashion. A binary or continuous time-frequency mask for resynthesis is inferred. Prominent separation models are factorial vector quantization [5] and FHMMs [6] which incorporate time dependencies. Some recent extensions of the FHMM framework have been proposed in [7], [8], [9]. Furthermore, non-negative matrix factorization (NMF) [10] is popular for identifying individual sources with temporal structure [11], [12].

Recently, an FHMM-based system has won the monaural speech separation and recognition challenge [13], [14]. This system includes speaker interaction models and temporal constraints. Remarkably, this model slightly outperforms human listeners on a restricted task [15]. Independently, we developed a similar FHMM model for multipitch tracking [16]. This model allows to infer the fundamental frequency contours of multiple concurrent speakers. Harmonic sounds such as voiced speech carry most of its energy at the fundamental frequency and integer multiples of it. Because a considerable fraction of speech consists of voiced phones, pitch is considered as important cue for speech segregation from mixtures (see e.g. [17], [18]). More general, many approaches in CASA make use of fundamental frequency estimates to perform perceptual grouping of harmonic sound sources in an acoustic scene [4]. For a comprehensive review on multipitch estimation, we refer the interested reader to [19], [20].

FHMM models are well-suited for modeling acoustic scenes of multiple concurrent sources. However, these models usually require speaker/source specific data for learning which limits the applicability. In this paper, we develop an algorithm for model adaptation to overcome any mismatch between training

Corresponding author, E-mail address: pernkopf@tugraz.at, Phone: +43 (0) 316 873 4436, Fax: +43 (0) 316 873 4432; This work was supported by the Austria sciene fund, project number P25244-N15 and S10610-N13.

and testing conditions. The aim is to adapt *universal* models learned on data from many speakers to novel environmental conditions, e.g. specific speakers, varying gain and/or acoustic environment, etc., using only speech mixture data. We propose an EM-based iterative algorithm using maximum likelihood linear regression (MLLR) for adaptation of speaker models from speech mixtures, and demonstrate multipitch tracking results obtained for gain adaptation and a distant talking scenario of two speakers which includes room reverberation. Our approach *does not* require clean source-specific data for adaptation. Furthermore, we constrain MLLR to modifications on the spectral envelope only. As we show, this is beneficial in cases of few adaptation data. Efficient inference is needed for model adaptation since the computational demands of exact inference in FHMMs scale exponentially with the number of sources. We propose a method for approximate inference based on pruning the state combinations of the observation likelihood. Our method is based on computationally efficient upper and lower bounds of the observation likelihood. We provide an evaluation in terms of accuracy and time requirements. This is in contrast to recently proposed approximate inference techniques based on loopy belief propagation, variational inference, and sampling [15], [21], [16].

The paper is organized as follows: In Section II we introduce FHMMs, the speaker interaction model, and provide a paragraph about model learning. In Section III and IV we present efficient inference methods based on likelihood pruning and the EM-framework for model adaptation, respectively. In Section V empirical results are reported and Section VI concludes the paper with a perspective on future work.

II. FHMMs with Probabilistic Interaction models

While HMMs usually model the speech of a single speaker, FHMMs are capable to model a mixture of several speakers as a joint effect of multiple Markov processes evolving in parallel over time, i.e. each Markov process models a particular source. For simplicity we consider the case of two interfering speakers throughout this section. This is generalized to K speakers subsequently. First proposed by Varga and Moore in the context of robust ASR [22], Ghahramani et al. [21] introduced FHMMs in a more general way together with novel mechanisms for inference and learning. By combining two single speaker HMMs using an interaction model, we obtain the FHMM shown in Figure 1 [16].[1]

The hidden random variables (RVs) $x_k^{(t)}$ represent the pitch state at time t of speaker (Markov chain) k. The discrete hidden variable $x_k^{(t)}$ has $|X| = 170$ states, where state value '1' refers to 'no pitch' (i.e. unvoiced speech or silence), and state values '2'-'170' encode different pitch frequencies ranging from 80 to 500Hz. As in [16], the pitch value of state $x \in \{2, ..., 170\}$ is $f_0 = \frac{16000}{30+x}$. Vector $\boldsymbol{s}_k^{(t)} \in \mathbb{R}^D$ corresponds to D bins of the short-time log-magnitude DFT of speaker k at time frame t. The dependency of $x_k^{(t)}$ over time is modeled by the transition probability $p(x_k^{(t)}|x_k^{(t-1)})$ and the prior distribution is denoted by $p(x_k^{(1)})$. The dependency of $\boldsymbol{s}_k^{(t)}$ on $x_k^{(t)}$ is modeled as Gaussian mixture model (GMM) according to[2]

$$p(\boldsymbol{s}_k|x_k) = p(\boldsymbol{s}_k|\boldsymbol{\Theta}_{k,x_k}) = \sum_{m=1}^{M_{k,x_k}} \alpha_{k,x_k}^m \mathcal{N}\left(\boldsymbol{s}_k|\boldsymbol{\theta}_{k,x_k}^m\right), \quad (1)$$

where $M_{k,x_k} \geq 1$ is the number of mixture components for speaker k and state x_k, and α_{k,x_k}^m corresponds to the weight of component m. These weights are constrained to be non-negative, $\alpha_{k,x_k}^m \geq 0$, and $\sum_{m=1}^{M_{k,x_k}} \alpha_{k,x_k}^m = 1$. The GMM for state x_k is fully specified by the parameter set $\boldsymbol{\Theta}_{k,x_k} = \left\{\alpha_{k,x_k}^m, \boldsymbol{\theta}_{k,x_k}^m\right\}_{m=1}^{M_{k,x_k}}$, where $\boldsymbol{\theta}_{k,x_k}^m = \left\{\boldsymbol{\mu}_{k,x_k}^m, \boldsymbol{\Sigma}_{k,x_k}^m\right\}$ is the mean and covariance of component m. We assume diagonal covariance matrices. To keep the notation compact, we use braces to denote a set of RVs from all Markov chains, e.g. $\{x_k^{(t)}\} := \{x_k^{(t)}\}_{k=1}^K$. At each time frame, the observation $\boldsymbol{y}^{(t)}$ is considered to be produced jointly by the two single-speaker emissions $\boldsymbol{s}_1^{(t)}$ and $\boldsymbol{s}_2^{(t)}$ using an interaction model $p(\boldsymbol{y}^{(t)}|\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)})$ presented in the sequel.

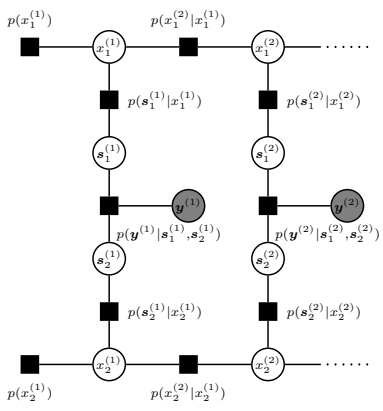

Fig. 1. FHMM represented as factor graph. Each Markov chain models the pitch trajectory of one speaker. At each time frame, the single-speaker emissions $\boldsymbol{s}_1^{(t)}$ and $\boldsymbol{s}_2^{(t)}$ jointly produce the observation $\boldsymbol{y}^{(t)}$. This process is modeled with the interaction model $p(\boldsymbol{y}^{(t)}|\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)})$.

A. The Mixture-Maximization Speaker Interaction Model

We use the mixture-maximization (MIXMAX) model originally proposed in [23], [22] for noise robust speech recognition.[3] It is based on the insight that the log-magnitude DFT of two speakers can be approximated by the element-wise maximum of their respective single-speaker log-magnitude DFT, i.e. $\boldsymbol{y}^{(t)} \approx \max(\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)})$. This approximation is based on the sparse nature of speech in time-frequency representations where each bin of a speech mixture spectrogram is dominated by a single speaker.

We obtain the pitch-conditional observation probability by

[1] Factor nodes are depicted as shaded rectangles together with their functional description. Hidden variable nodes are shown as white circles, observed variable nodes as gray circles.

[2] We omit the explicit dependence of random variables on t, where appropriate throughout the manuscript.

[3] Various types of interaction models are summarized in [24].

marginalization over $\boldsymbol{s}_k^{(t)}$, i.e.

$$p(\boldsymbol{y}^{(t)}|x_1^{(t)}, x_2^{(t)}) = \int\int p(\boldsymbol{y}^{(t)}|\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)}) p(\boldsymbol{s}_1^{(t)}|x_1^{(t)}) p(\boldsymbol{s}_2^{(t)}|x_2^{(t)}) d\boldsymbol{s}_1^{(t)} d\boldsymbol{s}_2^{(t)}. \quad (2)$$

This can be solved in closed form using single-speaker GMMs $p(\boldsymbol{s}_k^{(t)}|x_k^{(t)})$ and the MIXMAX interaction model[4] $p(\boldsymbol{y}^{(t)}|\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)}) = \delta\left(\boldsymbol{y}^{(t)} - \max(\boldsymbol{s}_1^{(t)}, \boldsymbol{s}_2^{(t)})\right)$, i.e.

$$p(\boldsymbol{y}^{(t)}|x_1, x_2) = \sum_{m_1=1}^{M_{1,x_1}} \sum_{m_2=1}^{M_{2,x_2}} \alpha_{1,x_1}^{m_1} \alpha_{2,x_2}^{m_2} \cdot \prod_{d=1}^{D} \left\{ \mathcal{N}(y_d^{(t)}|\theta_{1,x_1}^{m_1,d}) \Phi(y_d^{(t)}|\theta_{2,x_2}^{m_2,d}) + \Phi(y_d^{(t)}|\theta_{1,x_1}^{m_1,d}) \mathcal{N}(y_d^{(t)}|\theta_{2,x_2}^{m_2,d}) \right\}, \quad (3)$$

where y_d denotes the d^{th} element of $\boldsymbol{y}$, $\theta_{k,x_k}^{m_k,d}$ gives the d^{th} element of the corresponding mean and variance of the single-speaker model of speaker k, and $\Phi(y|\theta) := \int_{-\infty}^{y} \mathcal{N}(x|\theta)dx$ represents the univariate cumulative normal distribution (details are in [16], [23]).

B. Model Training

Having defined the structure of the FHMM, we have to determine $p(x_k^{(t)}|x_k^{(t-1)})$, $p(x_k^{(1)})$, and $p(\boldsymbol{s}_k|x_k)$. In general, we perform training using a set of pitch-labeled single-speaker utterances. Training can be performed either in a speaker dependent (SD) or in a speaker independent (SI) fashion. For SD training only speaker specific speech utterances are used, whereas for SI training utterances from a large amount of different speakers are required. Obviously, SD models yield better results, cf. [16], however, collecting sufficient data for SD models might be difficult or even impossible in practice.

The transition probabilities $p(x_k^{(t)}|x_k^{(t-1)})$ and prior distribution $p(x_k^{(1)})$ are obtained by maximum likelihood estimation using the reference pitch values from the single-speaker recordings. Additionally, Laplace smoothing[5] is applied. The parameters of the GMM $p(\boldsymbol{s}_k^{(t)}|x_k^{(t)})$ can be obtained by collecting all log-magnitude spectrogram frames of a particular reference pitch $x_k^{(t)}$ in the training set, and applying the EM-algorithm [25]. We use the minimum description length (MDL) criterion [26] to determine the number of components of each GMM automatically, where the maximal number of components per GMM was restricted to 20.

III. Likelihood Pruning for Efficient Inference

Given the observation sequence $\mathcal{Y} = \bigcup_{t=1}^{T} \boldsymbol{y}^{(t)}$, the aim of inference is to determine the instantiation of the hidden state sequences $\mathcal{X} = \bigcup_{t=1}^{T} \{x_k^{(t)}\}$ that maximizes the conditional distribution: $\mathcal{X}^* = \arg\max_{\mathcal{X}} p(\mathcal{X}|\mathcal{Y})$. This is also known as maximum a-posteriori inference and an *exact* solution is provided by the Viterbi algorithm [21], [16]. One limitation of exact inference in FHMMs is its high computational demand, i.e. for exact inference all possible combinations of pitch states across speakers need to be considered. Hence, the computational complexity scales exponentially with the number of Markov chains.

Approximate inference in FHMMs using Gibbs sampling, mean-field, or structured variational methods have been proposed in [21]. These approximate methods scale linearly with the number of sources using a linear interaction model. Among those methods, structured variational inference is favored; it is faster than Gibbs sampling and typically approximates the exact solution well. In [27], [14], variational inference in conjunction with loopy belief propagation is proposed. The posterior over state combinations is approximated using a set of variational distributions, which factorize across all simultaneous speakers. This is advantageous since the messages sent by loopy belief propagation across Markov chains can be approximated without the need to consider combinations of speaker states. The complexity of inference is linear in the number of speakers, however, messages passed between Markov chains need to be computed using an *iterative* scheme.

Here, we propose fast approximate inference for K simultaneous speakers based on likelihood pruning. Extending the model to K speakers, the pitch-conditional observation probability under the MIXMAX interaction model of (3) is given as

$$p(\boldsymbol{y}^{(t)}|\{x_k\}) = \sum_{\{m_k\}} \left(\prod_k \alpha_{k,x_k}^{m_k} \right) \prod_{d=1}^{D} \sum_k \mathcal{N}(y_d^{(t)}|\theta_{k,x_k}^{m_k,d}) \prod_{j \neq k} \Phi(y_d^{(t)}|\theta_{j,x_j}^{m_j,d}), \quad (4)$$

where $\sum_{\{m_k\}}$ refers to the nested sum $\sum_{m_1=1}^{M_{1,x_1}} \cdots \sum_{m_K=1}^{M_{K,x_K}}$. We make use of the fact that a large fraction of likelihood values is insignificantly small. We introduce novel upper and lower bounds on the state-conditional observation likelihood for efficiently retrieving a set of *probable* state configurations. The cardinality R of this set can be chosen to control the tradeoff between accuracy and computation time. Given a function $f(\cdot)$ defined over a finite domain $\mathcal{D}$, we define the *R-best set* of $f(\cdot)$, denoted by $\mathcal{S}_R^f$, as the set $\{x_r\}_{r=1}^R, x_r \in \mathcal{D}$, for which

$$f(x_r) \geq f(y) \qquad \forall x_r \in \mathcal{S}_R^f, \forall y \in \mathcal{D} \setminus \mathcal{S}_R^f. \quad (5)$$

The short-hand $f(\mathcal{S})$ denotes the set $\{f(x_r)\}_{r=1}^R$, where $\{x_r\}_{r=1}^R = \mathcal{S} \subseteq \mathcal{D}$.

For tracking, we use exact likelihood computation exclusively for the resulting R-best set of promising state combinations, where R is a small fraction of all $|X|^K$ likelihoods. Finally, a modified Viterbi algorithm for sparse lists of likelihoods is applied to determine the pitch trajectory of each speaker $\mathcal{X}^*$. As we show in the experiments (see Section V), the multi-pitch tracking performance of the pruning scheme is comparable to the results obtained with exact inference for small values of R. Moreover, tracking results for instantaneous mixtures of three speakers are presented in [24].

[4]Any deterministic function $\boldsymbol{y} = f(\boldsymbol{s}_1, \boldsymbol{s}_2)$ can be expressed as $p(\boldsymbol{y}|\boldsymbol{s}_1, \boldsymbol{s}_2) = \delta(\boldsymbol{y} - f(\boldsymbol{s}_1, \boldsymbol{s}_2))$, where $\delta(\cdot)$ is the Dirac-delta function.

[5]Laplace smoothing adds a one to each bin count of the discrete probability distribution before normalization. This avoids probability values of zero.

A. Computationally Efficient Bounds on the Likelihood

We rewrite Equation (4) as

$$p(\boldsymbol{y}|\{x_k\}) = \sum_{\{m_k\}} \left(\prod_k \alpha_{k,x_k}^{m_k} \right) L(\{x_k\},\{m_k\}), \qquad (6)$$

where we omit the dependency on t and introduce symbol L for the likelihood as a function of pitch and component states neglecting explicit dependency on $\boldsymbol{y}$, i.e.

$$\begin{aligned} L(\{x_k\},\{m_k\}) &= \prod_{d=1}^{D} \sum_k \mathcal{N}(y_d|\theta_{k,x_k}^{m_k,d}) \prod_{j\neq k} \Phi(y_d|\theta_{j,x_j}^{m_j,d}) \\ &= \prod_{d=1}^{D} \sum_k \mathcal{N}_{k,x_k}^{m_k,d} \prod_{j\neq k} \Phi_{j,x_j}^{m_j,d}. \end{aligned} \qquad (7)$$

As derived in Appendix A and B, the following upper and lower bounds hold for $\ln L(\cdot,\cdot)$:

$$\begin{aligned} \ln L(\{x_k\},\{m_k\}) &\leq \sum_k \underbrace{\sum_d \ln \left\{ \mathcal{N}_{k,x_k}^{m_k,d} + \Phi_{k,x_k}^{m_k,d} \right\}}_{=:u_k(x_k,m_k)} \\ &=: \mathrm{UB}(\{x_k\},\{m_k\}), \end{aligned} \qquad (8)$$

and

$$\begin{aligned} \ln L(\{x_k\},\{m_k\}) &\geq \max_k \left\{ \sum_d \ln \mathcal{N}_{k,x_k}^{m_k,d} + \sum_{j\neq k} \sum_d \ln \Phi_{j,x_j}^{m_j,d} \right\} \\ &=: \mathrm{LB}(\{x_k\},\{m_k\}). \end{aligned} \qquad (9)$$

Experiments on real data show that both bounds can be very conservative for some cases, while they can be tight for other cases (especially the lower bound). Nevertheless, they are useful to extract a significant amount of promising state combinations based on the R-best set of both bounds. The key to efficient inference is that the R-best set of both the upper and the lower bound, $\mathcal{S}_R^{\mathrm{UB}}$ and $\mathcal{S}_R^{\mathrm{LB}}$, can be calculated in a fast and efficient way for $R << \Lambda$, where Λ is the number of all possible combinations of pitch states and GMM components across speakers, i.e. $\Lambda = \sum_{\{x_k\}} \prod_k M_{k,x_k}$. To see this, consider the upper bound in (8). We exploit the fact that the upper bound is decomposable, i.e. each term $u_k(x_k,m_k)$ depends on one speaker only. From this it follows that $\mathcal{S}_R^{\mathrm{UB}}$ is guaranteed to be a subset of $\mathcal{S}_R^{u_1} \times \mathcal{S}_R^{u_2} \times \cdots \times \mathcal{S}_R^{u_K}$. Moreover, with high probability, $\mathcal{S}_R^{\mathrm{UB}}$ is in $\mathcal{S}_{\bar{R}}^{u_1} \times \mathcal{S}_{\bar{R}}^{u_2} \times \cdots \times \mathcal{S}_{\bar{R}}^{u_K}$ for some $\bar{R} < R$. An approximate but efficient way to obtain the R-best set of the upper bound is to determine the $\bar{R}$-best set of $u_k(x_k,m_k)$, $\mathcal{S}_{\bar{R}}^{u_k}$, for $k = 1,2,\ldots,K$, followed by explicit computation and sorting of all values of $\mathrm{UB}(\mathcal{S}_{\bar{R}}^{u_1} \times \mathcal{S}_{\bar{R}}^{u_2} \times \cdots \times \mathcal{S}_{\bar{R}}^{u_K})$. During preliminary experiments, we found that setting $\bar{R} = 3\sqrt{R}$ still works well. We argue that limiting the search to a reduced set of probable indices is a reasonable approximation with the benefit of heavily reduced computational efforts. A similar principle can be applied to efficiently compute the R-best set of the lower bound $\mathcal{S}_R^{\mathrm{LB}}$ in (9).

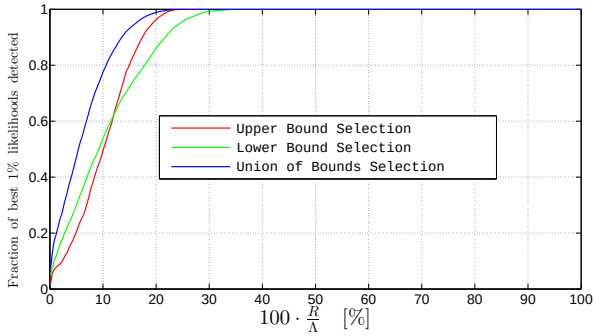

Fig. 2. Recall of the three selection strategies as a function of R, computed for a single time frame from a mixture of two speakers. On the x-axis, values of R are shown relative to the total number of likelihood elements.

B. State Selection Strategies Based on Likelihood Bounds

We propose three strategies to select the R-best set of indices:

1) Upper Bound Selection (UBS): Use $\mathcal{S}_R^{\mathrm{UB}}$.
2) Lower Bound Selection (LBS): Use $\mathcal{S}_R^{\mathrm{LB}}$.
3) Union of Bounds Selection (UNBS): Use the R-best set $\mathcal{S}_R^{\mathrm{UB}\cup\mathrm{LB}}$ of $\mathcal{S}_R^{\mathrm{UB}} \cup \mathcal{S}_R^{\mathrm{LB}}$.

We define the *recall* of a selection strategy in order to compare the three strategies in retrieving the best 1% likelihoods; specifically, the recall of the UBS method is

$$\mathrm{recall}_R^{\mathrm{UB}} = \frac{|\mathcal{S}_{\tilde{R}}^{LL} \cap \mathcal{S}_R^{\mathrm{UB}}|}{|\mathcal{S}_{\tilde{R}}^{LL}|}, \qquad (10)$$

where $\tilde{R} = \lfloor \frac{\Lambda}{100} \rfloor$ (i.e. 1% of the total number of likelihood elements), and $\mathcal{S}_{\tilde{R}}^{LL}$ is the $\tilde{R}$-best set of $L(\{x_k\},\{m_k\})$. The recall measures the fraction of 1% best likelihoods contained in $\mathcal{S}_R^{\mathrm{UB}}$. We define the recall of the LBS and UNBS analogously. Figure 2 shows the recall of $\mathcal{S}_R^{\mathrm{UB}}$, $\mathcal{S}_R^{\mathrm{LB}}$ and $\mathcal{S}_R^{\mathrm{UB}\cup\mathrm{LB}}$ as a function of R, evaluated on a representative frame from a mixture of two speakers. We see that UNBS works best, as it always retrieves the highest fraction of the best 1% likelihoods. The best 22% elements of Λ obtained with the UNBS method include all of the best 1% likelihoods. Results are similar for other frames.

Given the index set $\mathcal{S}$ containing the selected likelihood indices, we compute the *exact* observation probabilities (6) for tracking at states $\{x_k\} \in \mathcal{P}$, where $\mathcal{P} = \{\{x_k\}|\exists\{m_k\} : (\{x_k\},\{m_k\}) \in \mathcal{S}\}$ is the set of selected pitch combinations. All other observation probabilites remain zero. It is straightforward to extend the Viterbi algorithm presented in [21], [16] to sparse observation likelihood matrices (or tensors, for $K > 2$) for K Markov chains. In particular, the maximization in the recursion of Viterbi only needs to consider states $x_k^{(t)}$ that have been selected. Unfortunately, the computational complexity of this sparse Viterbi algorithm is still exponential in the number of Markov chains. In practice however, this variant is considerably faster compared to exact inference, and extends practical applicability to a higher number of Markov chains.

IV. Model Adaptation

We might encounter different channel conditions in the test case, i.e. the spectral characteristics of each source signal might have changed due to multi-path propagation in a room or a different microphone transfer function. The same holds for the problem of gain mismatches. Any mismatch between the speaker models and the actual condition in a recording results in a degraded tracking accuracy. The aim of model adaptation is to tune the available speaker models to the specific speaker characteristics and channel conditions that are present in a previously unseen recording using only the observed speech mixture.

A. Relevant Work

Some of the most successful approaches for model adaptation in the context of speech recognition are the MLLR framework [28], [29], maximum a posteriori (MAP) estimation [30], and rapid adaptation in eigenvoice space [31]. While these approaches assume that adaptation data consists of clean speech, methods for learning or adaptation of source models from contaminated speech have been developed as well. The aim is then to learn or adapt an *undistorted* source model from corrupted speech. One of the earliest approaches has been proposed in [23] where speech and noise – separately represented by individual models – are combined using the MIXMAX model. A mechanism is proposed to estimate the GMM-based speaker model from noisy speech assuming a known noise model. Rose et al. [32] extended this in terms of a more general interaction and background noise model based on GMMs. In [33], the eigenvoice approach is generalized to adapt individual speaker models given a superposition of two speech signals. Other approaches are summarized in detail in [24]. Furthermore, for the problem of gain adaptation from speech mixtures, methods based on iterated tracking and derivative-free optimization have been introduced in [34], [35]. In [36], [37], two different EM algorithms for gain adaptation are proposed.

B. Cepstrally Smoothed MLLR for FHMM Model Adaptation from Speech Mixtures

Similar to the original MLLR approach [28], the proposed adaptation method applies an affine transform to the mean parameters of the speaker model. For the multi-pitch tracking framework, only the spectral envelopes of a speaker model should be subject to adaptation, while all fine-spectral structure modeled by each pitch-conditional GMM should remain unmodified. Hence, changing vocal tract characteristics and channel conditions can be captured, while still ensuring that each GMM represents its associated pitch. For this reason, we propose an affine transform of the log-spectrum mean vectors which is implicitly constrained in cepstral domain.

1) Parameter Transform with Implicit Cepstral Smoothing: For simplicity, we assume that the mean parameters of all GMMs associated with speaker model k are subject to the same transform, i.e. we have full parameter tying across all GMMs and their components. However, this can be easily extended to the case of distinct transforms for subsets of GMMs. We propose the following transform for the mean of speaker k, state x_k and component m_k:

$$\hat{\boldsymbol{\mu}}_{k,x_k}^{m_k} = \boldsymbol{W}\left(\tilde{\boldsymbol{T}}_k \boldsymbol{W}\boldsymbol{\mu}_{k,x_k}^{m_k} + \tilde{\boldsymbol{b}}_k\right) = \boldsymbol{W}\tilde{\boldsymbol{T}}_k\boldsymbol{W}\boldsymbol{\mu}_{k,x_k}^{m_k} + \boldsymbol{W}\tilde{\boldsymbol{b}}_k,$$

where matrix $\boldsymbol{W}$ denotes the $D \times D$ discrete cosine transform (DCT) (type I) matrix:

$$W_{i,j} = \begin{cases} \frac{1}{\sqrt{2D-2}} \cos\left(\frac{\pi}{D-1}(i-1)(j-1)\right) & \text{if } j \in \{1, D\}, \\ \frac{2}{\sqrt{2D-2}} \cos\left(\frac{\pi}{D-1}(i-1)(j-1)\right) & \text{otherwise,} \end{cases} \tag{11}$$

which essentially maps a mean vector $\boldsymbol{\mu}$ from log-spectral to cepstral domain.[6] The affine transform of the cepstral representation $\tilde{\boldsymbol{T}}_k\boldsymbol{W}\boldsymbol{\mu}_{k,x_k}^{m_k} + \tilde{\boldsymbol{b}}_k$ is back-transformed by $\boldsymbol{W}$ to log-spectral domain $\hat{\boldsymbol{\mu}}_{k,x_k}^{m_k}$.[7]
We constrain the structure of the cepstral transform matrix to the form

$$\tilde{\boldsymbol{T}}_k = \left(\begin{array}{c|c} \boldsymbol{T}_k & \boldsymbol{0} \\ \hline \boldsymbol{0}^T & \boldsymbol{I} \end{array}\right), \tag{12}$$

where we denote by $\boldsymbol{T}_k$ a $C \times C$ submatrix, $\boldsymbol{I}$ is the $(D-C)\times(D-C)$ identity matrix and $\boldsymbol{0}$ the $C \times (D-C)$ zero matrix. We constrain the bias vector $\tilde{\boldsymbol{b}}_k = \left(\frac{\boldsymbol{b}_k}{\boldsymbol{0}}\right)$ likewise. As a result, only the first C low-order coefficients of the cepstral representation of $\boldsymbol{\mu}_k$ are subject to the affine transform defined by $\boldsymbol{T}_k$ and $\boldsymbol{b}_k$. Setting $C = 1$ is sufficient to perform gain adaptation.[8] For $C = D$, no constraints are imposed on the transform, and the method is conceptually equivalent to MLLR (apart from the fact that adaptation parameters are defined in cepstral domain). For small amounts of adaptation data, choosing some $C < D$ can help to avoid overfitting. We refer to this method as *cepstrally smoothed* MLLR (csMLLR).

2) General EM Algorithm for MLLR-Based FHMM Model Adaptation from Speech Mixtures: Here, we derive update equations to learn transformation parameters for each individual speaker model, $\boldsymbol{T}_k$ and $\boldsymbol{b}_k$, given a mixture of speech. We adapt parameters by maximizing the log-likelihood given the observed speech mixture:

$$\mathrm{LL}(\{\boldsymbol{T}_k\}, \{\boldsymbol{b}_k\}) = \ln \sum_{\mathcal{X}} p(\mathcal{X}, \mathcal{Y} | \{\boldsymbol{T}_k\}, \{\boldsymbol{b}_k\}), \tag{13}$$

where

$$p(\mathcal{X}, \mathcal{Y} | \{\boldsymbol{T}_k\}, \{\boldsymbol{b}_k\}) = \tag{14}$$
$$\prod_{k=1}^{K} \left[p(x_k^{(1)}) \prod_{t=2}^{T} p(x_k^{(t)} | x_k^{(t-1)}) \right] \prod_{t=1}^{T} p(\boldsymbol{y}^{(t)} | \{x_k^{(t)}\}, \{\boldsymbol{T}_k\}, \{\boldsymbol{b}_k\})$$

is the joint distribution of all observed data and hidden variables of an FHMM with K Markov chains. The conditional dependency on the transformation parameters is made explicit.

[6] The GMM-based speaker model introduced in Section II does not include the spectral energy at zero Hz, as it does not hold any pitch-related information. In this section, however, we do assume that $\boldsymbol{\mu}$ additionally contains the bias bin at zero Hz, because it simplifies the application of the DCT transform and all related notation.

[7] Note that matrix $\boldsymbol{W}$ is an involution, i.e. $\boldsymbol{W}\boldsymbol{W} = \boldsymbol{I}$.

[8] Note that the gain is *additive* in log-spectrum/cepstrum domain.

The distribution of the observation at one time instance given the hidden pitch states is (cf. (2))

$$p(\boldsymbol{y}^{(t)}|\{x_k^{(t)}\},\{\boldsymbol{T}_k\},\{\boldsymbol{b}_k\}) = \tag{15}$$
$$\int\!\cdots\!\int p(\boldsymbol{y}^{(t)}|\{\boldsymbol{s}_k^{(t)}\})\prod_{k=1}^{K} p(\boldsymbol{s}_k^{(t)}|x_k^{(t)},\boldsymbol{T}_k,\boldsymbol{b}_k)d\boldsymbol{s}_1^{(t)}\cdots d\boldsymbol{s}_K^{(t)}$$
$$= \sum_{\{m_k\}}\prod_{k=1}^{K}\alpha_{k,x_k}^{m_k}\int\!\cdots\!\int p(\boldsymbol{y}^{(t)}|\{\boldsymbol{s}_k^{(t)}\})$$
$$\times\prod_{k=1}^{K} p(\boldsymbol{s}_k^{(t)}|x_k^{(t)},m_k,\boldsymbol{T}_k,\boldsymbol{b}_k)d\boldsymbol{s}_1^{(t)}\cdots d\boldsymbol{s}_K^{(t)}, \tag{16}$$

where no explicit assumption has been made on the interaction model.

It is difficult to maximize the log-likelihood in (13) directly. Instead, Jensen's inequality is applied to construct a lower bound on (13), which is in general easier to optimize [38]. For any distribution $q(\cdot)$, and any joint probability $p(\mathcal{X},\mathcal{Y})$, it follows from Jensen's inequality that

$$\ln\sum_{\mathcal{X}} p(\mathcal{X},\mathcal{Y}) = \ln\sum_{\mathcal{X}} q(\mathcal{X})\frac{p(\mathcal{X},\mathcal{Y})}{q(\mathcal{X})} \geq \sum_{\mathcal{X}} q(\mathcal{X})\ln\frac{p(\mathcal{X},\mathcal{Y})}{q(\mathcal{X})},$$

and equality holds if and only if $q(\mathcal{X}) = p(\mathcal{X}|\mathcal{Y})$. We systematically apply Jensen's inequality to construct the following sequence of variational lower bounds on the LL in (13):

$$\text{LL} \geq \text{const} + \sum_{\mathcal{X}} q(\mathcal{X})\ln p(\mathcal{X},\mathcal{Y}|\{\boldsymbol{T}_k\},\{\boldsymbol{b}_k\}) \tag{17}$$
$$= \text{const} + \sum_{\mathcal{X}} q(\mathcal{X})\sum_{t=1}^{T}\ln p(\boldsymbol{y}^{(t)}|\{x_k^{(t)}\},\{\boldsymbol{T}_k\},\{\boldsymbol{b}_k\}) \tag{18}$$
$$\geq \text{const} + \sum_{\mathcal{X}} q(\mathcal{X})\sum_{t=1}^{T}\sum_{\{m_k\}} q(\{m_k\})$$
$$\times\ln\int\!\cdots\!\int p(\boldsymbol{y}^{(t)}|\{\boldsymbol{s}_k^{(t)}\})\prod_{k=1}^{K} p(\boldsymbol{s}_k^{(t)}|x_k^{(t)},m_k,\boldsymbol{T}_k,\boldsymbol{b}_k)d\boldsymbol{s}_1^{(t)}\cdots d\boldsymbol{s}_K^{(t)} \tag{19}$$
$$\geq \text{const} + \sum_{\mathcal{X}} q(\mathcal{X})\sum_{t=1}^{T}\sum_{\{m_k\}} q(\{m_k\})$$
$$\times\int\!\cdots\!\int q(\{\boldsymbol{s}_k^{(t)}\})\sum_{k=1}^{K}\ln p(\boldsymbol{s}_k^{(t)}|x_k^{(t)},m_k,\boldsymbol{T}_k,\boldsymbol{b}_k)d\boldsymbol{s}_1^{(t)}\cdots d\boldsymbol{s}_K^{(t)}, \tag{20}$$

where const refers to all terms independent of $\{\boldsymbol{T}_k\}$ and $\{\boldsymbol{b}_k\}$. Note that this lower bound is valid for an arbitrary choice of the variational distributions $q(\mathcal{X})$, $q(\{m_k\})$ and $q(\{\boldsymbol{s}_k^{(t)}\})$. Starting with an initial guess for the adaptation parameters, a local maximum of (13) can be found using the EM algorithm.

E-Step: The variational distributions are set such that the lower bound is tight[9] at the current parameter estimate, i.e.

$$q(\{x_k^{(t)}\}) = \sum_{\mathcal{X}\setminus\{x_k^{(t)}\}} p(\mathcal{X}|\mathcal{Y},\{\boldsymbol{T}_k^{(old)}\},\{\boldsymbol{b}_k^{(old)}\}), \tag{21}$$
$$q(\{m_k\}) = p(\{m_k\}|\boldsymbol{y}^{(t)},\{x_k^{(t)}\},\{\boldsymbol{T}_k^{(old)}\},\{\boldsymbol{b}_k^{(old)}\}), \text{and} \tag{22}$$
$$q(\{\boldsymbol{s}_k^{(t)}\}) = p(\{\boldsymbol{s}_k^{(t)}\}|\boldsymbol{y}^{(t)},\{x_k^{(t)}\},\{m_k\},\{\boldsymbol{T}_k^{(old)}\},\{\boldsymbol{b}_k^{(old)}\}). \tag{23}$$

Note that the calculation of Equations (21) and (22) during the E-Step is equivalent to the E-Step for exact parameter learning in FHMMs [21]. Specifically, (21) represents the marginal posterior, which can be obtained using the forward-backward algorithm of FHMMs as proposed in [21].

M-Step: The lower bound of the LL in (20) is maximized with respect to the parameters. For each speaker k, the parameters are updated according to

$$\{\boldsymbol{T}_k,\boldsymbol{b}_k\} = \arg\max_{\{\boldsymbol{T},\boldsymbol{b}\}} Q_k(\boldsymbol{T},\boldsymbol{b})$$
$$= \arg\max_{\{\boldsymbol{T},\boldsymbol{b}\}}\sum_{t,\{x_k^{(t)}\},\{m_k\}}\gamma_t(\{x_k^{(t)}\},\{m_k\})$$
$$\times\,\mathrm{E}_{\{\boldsymbol{s}_k^{(t)}\}}\Big\{\ln p(\boldsymbol{s}_k^{(t)}|x_k^{(t)},m_k,\boldsymbol{T},\boldsymbol{b})\Big\}, \tag{24}$$

where we introduced the shorthand

$$\gamma_t(\{x_k\},\{m_k\}) = p(\{x_k^{(t)}\},\{m_k\},|\mathcal{Y},\{\boldsymbol{T}_k^{(old)}\},\{\boldsymbol{b}_k^{(old)}\}) \tag{25}$$
$$= q(\{x_k^{(t)}\})q(\{m_k\}) \tag{26}$$

to denote the posterior of states and components obtained in the previous E-Step, i.e. Eqn. (21) and (22)). The objective (24) for the M-Step (commonly referred as auxiliary function) was obtained by plugging Equations (21), (22) and (23) into (20). The unknown single-speaker spectrum $\boldsymbol{s}_k^{(t)}$ has been replaced by its conditional expected value where the expectation $\mathrm{E}_{\{\boldsymbol{s}_k^{(t)}\}}\{\cdot\}$ is with respect to the distribution in (23). The relevant steps of this algorithm can be summarized as follows: During the E-Step, the expectation of the single speaker spectrum is estimated from the speech mixture, using the current estimate of all speaker models. During the M-Step, the expected single speaker spectrum is used as a surrogate for the unknown single speaker spectrum, and the adaptation parameters are estimated independently for each speaker.

3) EM Algorithm for Cepstrally-Smoothed MLLR-Based FHMM Model Adaptation from Speech Mixtures Using the MIXMAX Interaction Model: So far, we have not made an explicit assumption about the interaction model and on how exactly the transformation parameters $\{\boldsymbol{T}_k,\boldsymbol{b}_k\}$ enter the single-speaker model. Based on the general EM framework presented in the previous section, here we introduce the MIXMAX interaction model and the transformation for the GMM-based speaker model defined in (11), and derive the update equations for the M-Step of the EM algorithm. Consequently,

[9]Up to a term that does not depend on adaptation parameters.

each mixture component is of the form

$$p(\boldsymbol{s}_k^{(t)}|x_k^{(t)}, m_k, \boldsymbol{T}_k, \boldsymbol{b}_k) = \mathcal{N}(\boldsymbol{s}_k^{(t)}|\boldsymbol{W}\tilde{\boldsymbol{T}}_k\boldsymbol{W}\boldsymbol{\mu}^{m_k}_{k,x_k^{(t)}} + \boldsymbol{W}\tilde{\boldsymbol{b}}_k, \boldsymbol{\Sigma}^{m_k}_{k,x_k^{(t)}}). \quad (27)$$

Consider the auxiliary function $Q_k(\boldsymbol{T}_k, \boldsymbol{b}_k)$ for speaker k in (24). Plugging (27) into $Q_k(\boldsymbol{T}_k, \boldsymbol{b}_k)$ results in

$$Q_k(\boldsymbol{T}_k, \boldsymbol{b}_k) = \sum_{t,\{x_k\},\{m_k\}} \gamma_t(\{x_k\},\{m_k\}) \, \mathrm{E}_{\{\boldsymbol{s}_k^{(t)}\}}\Big\{\ln \mathcal{N}(\boldsymbol{s}_k^{(t)}|\boldsymbol{W}\tilde{\boldsymbol{T}}_k\boldsymbol{W}\boldsymbol{\mu}^{m_k}_{k,x_k} + \boldsymbol{W}\tilde{\boldsymbol{b}}_k, \boldsymbol{\Sigma}^{m_k}_{k,x_k})\Big\}. \quad (28)$$

As $Q_k(\cdot,\cdot)$ is jointly concave in $\boldsymbol{T}_k$ and $\boldsymbol{b}_k$, a global optimum can be obtained by setting the derivative to zero [39]. This leads to two conditions:
Condition (i): $\frac{\partial Q_k(\boldsymbol{T}_k,\boldsymbol{b}_k)}{\partial \boldsymbol{T}_k} = 0$, and
Condition (ii): $\frac{\partial Q_k(\boldsymbol{T}_k,\boldsymbol{b}_k)}{\partial \boldsymbol{b}_k} = 0$.

To facilitate the derivation of $Q_k(\cdot,\cdot)$, we define the following submatrices of the cosine transform matrix:

$$\hat{\boldsymbol{W}} = \boldsymbol{W}_{1:D,1:C}, \quad (29)$$
$$\check{\boldsymbol{W}} = \boldsymbol{W}_{1:C,1:D}, \text{ and} \quad (30)$$
$$\bar{\boldsymbol{W}} = \boldsymbol{W}_{1:D,(C+1):D}\boldsymbol{W}_{(C+1):D,1:D}, \quad (31)$$

where $\boldsymbol{W}_{1:D,(C+1):D}$ denotes the matrix containing the first D rows and the $(C+1)^{\text{th}}$ to the D^{th} column of $\boldsymbol{W}$. This way, we can re-express the linear transform as well as the bias vector in terms of the parameters $\boldsymbol{T}_k$ and $\boldsymbol{b}_k$ subject to optimization as

$$\boldsymbol{W}\tilde{\boldsymbol{T}}_k\boldsymbol{W} = \hat{\boldsymbol{W}}\boldsymbol{T}_k\check{\boldsymbol{W}} + \bar{\boldsymbol{W}}, \text{ and} \quad (32)$$
$$\boldsymbol{W}\tilde{\boldsymbol{b}}_k = \hat{\boldsymbol{W}}\boldsymbol{b}_k. \quad (33)$$

Condition (i): Setting $\frac{\partial Q_k(\boldsymbol{T}_k,\boldsymbol{b}_k)}{\partial \boldsymbol{T}_k} = 0$; it follows from standard matrix calculus [40] (see [24] for a derivation) that

$$\sum_{x_k,m_k} \gamma_{x_k,m_k}\boldsymbol{A}_{x_k,m_k}\boldsymbol{T}_k\boldsymbol{B}_{x_k,m_k} = \boldsymbol{C}, \quad (34)$$

where

$$\gamma_{x_k,m_k} = \sum_{\{x_j\}_{j\neq k}}\sum_{\{m_j\}_{j\neq k}}\sum_t \gamma_t(\{x_j\},\{m_j\}), \quad (35)$$
$$\boldsymbol{A}_{x_k,m_k} = \hat{\boldsymbol{W}}^T{\boldsymbol{\Sigma}^{m_k}_{k,x_k}}^{-1}\hat{\boldsymbol{W}}, \quad (36)$$
$$\boldsymbol{B}_{x_k,m_k} = \check{\boldsymbol{W}}\boldsymbol{\mu}^{m_k}_{k,x_k}{\boldsymbol{\mu}^{m_k}_{k,x_k}}^T\check{\boldsymbol{W}}^T, \quad (37)$$
$$\boldsymbol{C} = \sum_{x_k,m_k}\hat{\boldsymbol{W}}^T{\boldsymbol{\Sigma}^{m_k}_{k,x_k}}^{-1} \times\Big(\mathrm{E}\{\boldsymbol{s}_k|\mathcal{Y},x_k,m_k\} - \gamma_{x_k,m_k}\left(\bar{\boldsymbol{W}}\boldsymbol{\mu}^{m_k}_{k,x_k} + \hat{\boldsymbol{W}}\boldsymbol{b}_k\right)\Big){\boldsymbol{\mu}^{m_k}_{k,x_k}}^T\check{\boldsymbol{W}}^T \quad (38)$$

In Equation (35), the marginal posterior obtained during the E-Step is accumulated for all time frames and all states of concurrent speakers $j \neq k$. Similarly, the state-conditional expected single-speaker spectrum of speaker k is obtained by weighted accumulation:

$$\mathrm{E}\{\boldsymbol{s}_k|\mathcal{Y},x_k,m_k\} = \sum_{\{x_j\}_{j\neq k}}\sum_{\{m_j\}_{j\neq k}}\sum_t \gamma_t(\{x_j\},\{m_j\})\mathrm{E}\Big\{\boldsymbol{s}_k|\boldsymbol{y}^{(t)},\{x_k\},\{m_k\}\Big\}, \quad (39)$$

where $\mathrm{E}\big\{\boldsymbol{s}_k|\boldsymbol{y}^{(t)},\{x_k\},\{m_k\}\big\}$ is the expected single-speaker spectrum conditioned on the observation at time t as well as a pitch and component combination. The d^{th} dimension of this expectation is calculated as

$$\mathrm{E}\Big\{s_k^d|\boldsymbol{y}^{(t)},\{x_k\},\{m_k\}\Big\} = \frac{y_d^{(t)}\Psi^{m_k,d}_{k,x_k} + \left(\mu^{m_k,d}_{k,x_k} - (\sigma^{m_k,d}_{k,x_k})^2\Psi^{m_k,d}_{k,x_k}\right)\sum_{l\neq k}\Psi^{m_l,d}_{l,x_l}}{\sum_j \Psi^{m_j,d}_{j,x_j}}, \quad (40)$$

where

$$\Psi^{m_k,d}_{k,x_k} = \frac{\mathcal{N}(y_d|\theta^{m_k,d}_{k,x_k})}{\Phi(y_d|\theta^{m_k,d}_{k,x_k})} \quad (41)$$

is the ratio of the normal density and the cumulative normal distribution of observation y_d. For a derivation of (40), we refer the reader to [24]. Note that the calculation of the sufficient statistics (35) and (39) is intractable.
Assuming that $\boldsymbol{b}_k$ is fixed, the matrix equation in (34) can be solved in closed form [41] as

$$\mathrm{vec}(\boldsymbol{T}_k) = \left(\sum_{x_k,m_k}\gamma_{x_k,m_k}{\boldsymbol{B}_{x_k,m_k}}^T \otimes \boldsymbol{A}_{x_k,m_k}\right)^{-1}\mathrm{vec}(\boldsymbol{C}), \quad (42)$$

where $\otimes$ denotes the Kronecker product and $\mathrm{vec}(\boldsymbol{T})$ is a vector obtained by sequentially stacking the columns of $\boldsymbol{T}$.[10]
Condition (ii): Setting $\frac{\partial Q_k(\boldsymbol{T}_k,\boldsymbol{b}_k)}{\partial \boldsymbol{b}_k} = 0$; we obtain

$$\boldsymbol{b}_k = \left(\sum_{x_k,m_k}\gamma_{x_k,m_k}\hat{\boldsymbol{W}}^T{\boldsymbol{\Sigma}^{m_k}_{k,x_k}}^{-1}\hat{\boldsymbol{W}}\right)^{-1} \times\sum_{x_k,m_k}\hat{\boldsymbol{W}}^T{\boldsymbol{\Sigma}^{m_k}_{k,x_k}}^{-1}\Big(\mathrm{E}\{\boldsymbol{s}_k|\mathcal{Y},x_k,m_k\} - \gamma_{x_k,m_k}\boldsymbol{W}\tilde{\boldsymbol{T}}_k\boldsymbol{W}\boldsymbol{\mu}^{m_k}_{k,x_k}\Big), \quad (43)$$

and we can solve for $\boldsymbol{b}_k$ assuming that $\boldsymbol{T}_k$ is fixed.

Equations (42) and (43) are applied iteratively during the M-Step to update the value of one variable while holding the other fixed. This type of block-coordinate ascent method is guaranteed to converge to the global optimum, as the objective in (28) is jointly concave in $\boldsymbol{T}_k$ and $\boldsymbol{b}_k$.

The EM algorithm is summarized in detail in [24]. During the E-Step, the unknown single-speaker spectrum of every speaker is inferred, based on the currently available speaker models. During the M-Step, the expected single-speaker spectrum is used as a surrogate to the true single-speaker spectrum, and model parameters $\boldsymbol{T}_k$ and $\boldsymbol{b}_k$ are updated according to csMLLR. For the special case where only one speaker model is adapted from speech data of a single speaker, the E-Step is not necessary and the true single-speaker spectrum can be used in place of the expected speaker spectrum. Unfortunately, the forward-backward algorithm as well as the calculation of sufficient statistics during the E-Step of the exact algorithm are intractable. Therefore, we make use of the fast pruning scheme developed for the MIXMAX interaction model (see

[10] Using the Kronecker product, a product of three matrices $\boldsymbol{ATB}$ can be re-expressed as $\mathrm{vec}(\boldsymbol{ATB}) = (\boldsymbol{B}^T \otimes \boldsymbol{A})\mathrm{vec}(\boldsymbol{T})$ [41]. More details are provided in [24].

Section III). During the E-Step, the observation likelihoods are evaluated only for the R-best set of state combinations and then passed to a sparse variant of the forward-backward algorithm, which is based on the same ideas as the sparse Viterbi algorithm. This pruning scheme allows for a natural tradeoff between approximation quality and computational complexity, which can be directly controlled by the value of R. Another beneficial aspect is that the posterior is approximated in a single pass of the forward-backward algorithm. This is in contrast to variational methods, where the posterior is obtained by an iterative scheme. The pruning approach can be seen as a combination of the "sparse"- and "winner-take-all" variant of EM [42].

V. Experiments

In this section, we provide empirical results for both efficient inference (see Section V-C) and model adaptation (see Section V-D) using speech mixtures of two speakers. We start with a description of the experimental setup.

A. Data and Feature Extraction

For all experimental evaluations[11], we used material from the GRID corpus [44]. We selected three female (speaker 1, 2, and 3) and three male speakers (speaker 18, 20, and 21) as test speakers. For each test speaker, 450 sentences were used to train SD GMMs, 40 sentences were reserved as development data, and 3 sentences were used as test data. In addition to SD GMMs, SI GMMs were trained using female speakers 4, 7, 11, 15, 16, 22, 23, 24, 25, 29, 31, 33, 34 and male speakers 5, 6, 9, 10, 12, 13, 14, 17, 19, 26, 27, 28, 32. Again 450 sentences per speaker were used. The reference pitch trajectories needed for training and evaluation were obtained using the RAPT method [45].[12] No manual correction of the extracted reference pitch was performed. Hence, the ground truth may still contain some errors.

We create 135 test mixtures using test sentences of the 6 test speakers. Combining every test speaker with every other speaker results in 15 speaker pairs, and 9 test mixtures were created for each speaker pair. Mixing was performed by linear superposition (instantaneous mixture) and equal gain (except for the gain adaptation experiment in Section V-D2).

The observed features $\boldsymbol{y}^{(t)}$ are based on the log-spectrogram of the speech mixtures. The spectrogram is computed via the 1024 point DFT, using a Hamming window of length 32ms and step size of 10ms. The sampling frequency is $f_s = 16$kHz. Each observation vector $\boldsymbol{y}^{(t)} \in \mathbb{R}^{64}$ is obtained by taking the log-magnitude of spectral bins 2-65, which corresponds to a frequency range up to 1000Hz.

B. Performance Measures

For every test instance, each method estimates two pitch trajectories, $\tilde{f}_0^{(1)}[t]$ and $\tilde{f}_0^{(2)}[t]$. We slightly extended the error measures in [46] to evaluate the pitch-tracking performance in terms of successful speaker assignment. Let E_{ij} denote the percentage of time frames where i pitch points are misclassified as j pitch points, i.e. E_{12} means the percentage of frames with two pitch values estimated whereas only one pitch point is present. Each of the two estimated pitch trajectories are *globally* assigned to a ground truth trajectory, $f_0^{(1)}[t]$ or $f_0^{(2)}[t]$ based on the mean square error. We define the *speaker assigned pitch frequency deviation* as

$$\Delta f^{(k)}[t] = \frac{\left|\tilde{f}_0^{(k)}[t] - f_0^{(k)}[t]\right|}{f_0^{(k)}[t]}, \qquad (44)$$

[11] Further experiments for the PTDB-TUG corpus [43] are provided in [24]. The flavor of these experiments is highly similar to the presented results.

[12] An implementation of the RAPT algorithm is provided by the Entropic speech processing system (ESPS) labeled as "get_f0" method.

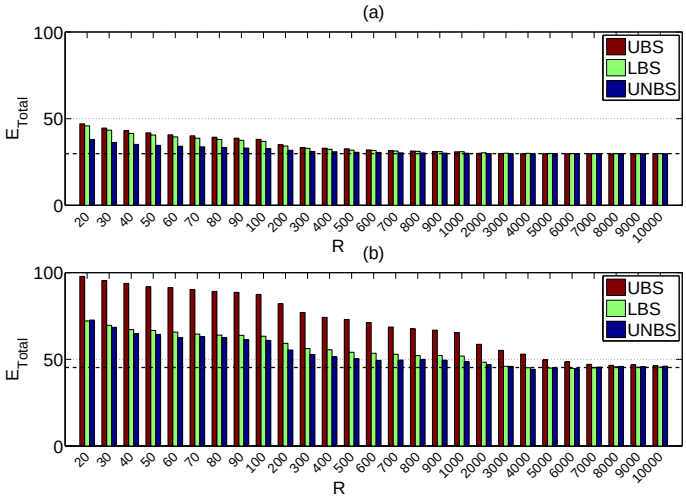

Fig. 3. Error measure E_{Total} for different settings of pruning parameter R: (a) SD models. (b) SI models. For each R, the mean performance is shown for the upper bound selection (UBS), lower bound selection (LBS) and the union of bounds selection (UNBS) method. The dashed horizontal line indicates the exact inference performance.

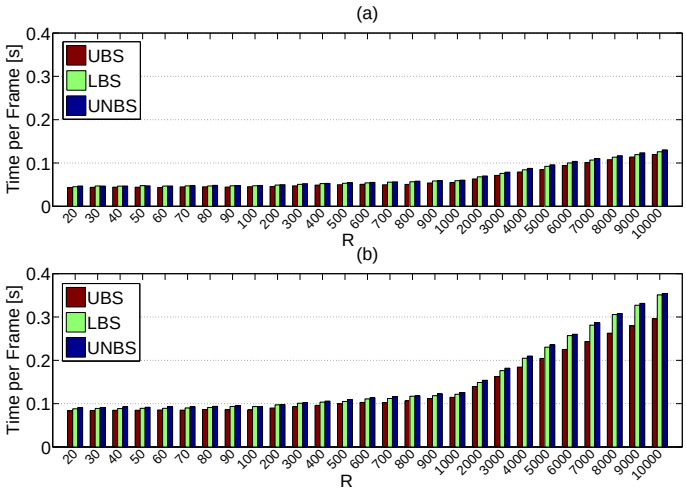

Fig. 4. Computation time of likelihood calculation per analysis frame, averaged on 6 test mixtures for different settings of pruning parameter R: (a) SD models. (b) SI models. For each R, the mean time in seconds is shown for the upper bound selection (UBS), lower bound selection (LBS) and the union of bounds selection (UNBS) method.

where $f_0^{(k)}[t]$ denotes the reference chosen for $\tilde{f}_0^{(k)}[t]$. For each reference trajectory, we define the corresponding permutation error $E_{Perm}^k[t]$ to be one at time frames where the voicing decision for both estimates is correct, but the pitch frequency deviation exceeds 20%, and $\tilde{f}_0^k[t]$ is within the 20% error bound of the other reference pitch. This indicates a permutation of pitch estimates due to incorrect speaker assignment. The overall permutation error rate E_{Perm} is the percentage of time frames where either $E_{Perm}^1[t]$ or $E_{Perm}^2[t]$ is one. Next, we define for each reference trajectory the corresponding gross error $E_{Gross}^k[t]$ to be one at time frames where the voicing decision is correct, but the pitch frequency deviation exceeds 20% and no permutation error was detected. The overall gross error rate E_{Gross} is the percentage of time frames where either $E_{Gross}^{(1)}[t]$ or $E_{Gross}^{(2)}[t]$ is one. The fine detection error $E_{Fine}^{(k)}$ is the average speaker assigned frequency deviation in percent at time frames where $\Delta f^{(k)}[t]$ is smaller than 20%. The overall error, E_{Total}, is $E_{Total} = E_{01}+E_{02}+E_{10}+E_{12}+E_{20}+E_{21}+E_{Gross}+E_{Fine}+E_{Perm}$, where $E_{Fine} = E_{Fine}^{(1)} + E_{Fine}^{(2)}$.

C. Results for Efficient Inference

We compare the performance of the three proposed bound selection methods on mixtures of two speakers.[13] We varied the pruning parameter R over a range of 20 to 10000. For each setting of R, we evaluated the performance E_{Total} of each of the three bound selection methods on 135 test mixtures. Results are shown in Figure 3. The dashed horizontal line shows the performance achieved by exact inference, i.e. without likelihood pruning. Each of the three bound selection methods approaches the performance of exact inference with increasing R. UNBS approaches this limit for the smallest R in all cases, while UBS performs worst.

To indicate the computation time of the methods involved, measurements were performed on a 3.2-GHz six core machine with 12-GB main memory. All algorithms were implemented and tested in Matlab. For the exact computation of the likelihoods for the selected pitch combinations $\mathcal{P}$, a Matlab-MEX implementation was used. Measurements were performed on six test mixtures of the GRID database, where only the time needed for likelihood pruning and subsequent calculation of selected likelihood values was measured. The measured time was normalized by the total number of frames T of the mixture. The averaged results are shown in Figure 4 for SD and SI models. In comparison, the average time per analysis frame needed for exact likelihood computation is 0.21 s when using SD models and 2.17 s when using SI models. Note that the SI model uses GMMs with more GMM components which explains the larger computation time. For small values of R, a constant computational overhead dominates the required time, while for larger increasing values of R the required time scales linearly with R. The UBS method is always fastest, however LBS and UNBS do not need excessively more time compared to UBS.

[13]Tracking results for three speakers are provided in [24].

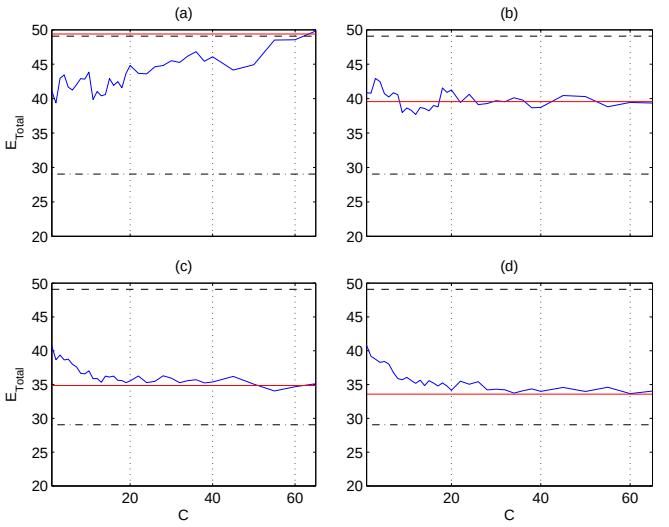

Fig. 5. Results for single-speaker adaptation. SI models where adapted on single-speaker data, using MLLR or csMLLR for different settings of C. Dashed solid line: SI model performance. Dash-dotted line: SD model performance. Blue solid line: csMLLR performance. Red solid line: MLLR performance. A different amount of adaptation data was used: (a) 1.8 s, (b) 3.5 s, (c) 8.7 s, (d) 17.5 s.

D. Results for Model Adaptation

We evaluate the capabilities of model adaptation for a range of scenarios. First, in Section V-D1, we assume that single-speaker recordings are available for SI model adaptation. Next, in Section V-D2, we evaluate the performance of gain adaptation given speech mixtures of unknown mixing level. In Section V-D3, we demonstrate results for model adaptation from speech mixtures recorded in a real office environment, i.e. SD models of both speakers are adapted on the test mixture itself.

1) Model Adaptation from Single-Speaker Data: We evaluated the multi-pitch tracking performance using SI models adapted on single-speaker utterances, i.e. we used adaptation utterances from the same speakers that are present in the test mixture. Experiments were performed using a subset of 4 (out of 6) randomly selected test speakers (two male and two female) with three test utterances each. This results in a total of 54 test mixtures. For adaptation, we used 40 utterances per test speaker from the development set. Note that no additional reference pitch labels are required for adaptation. SI models were adapted using different amounts of development data and using either MLLR or csMLLR with different settings of the smoothing parameter C.

Performance results in terms of E_{Total} are shown in Figure 5, together with the performance achieved using either SI or SD models. For small amounts of adaptation data (about 1.8 s), MLLR cannot improve performance over the SI model, while csMLLR achieves better performance for low values of C (strong smoothing). With larger amounts of adaptation data available, performance of MLLR steadily increases and settles at $E_{Total} \approx 34$ for 17.5 s of adaptation data – more adaptation data cannot further increase the performance due to the limited expressive power of a single shared affine transform (parameter tying). The results indicate that performance of

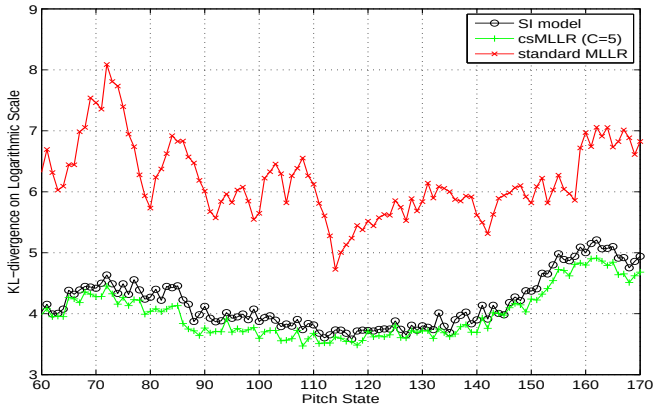

Fig. 6. For a range of relevant pitch states, the similarity between two pitch-conditional GMMs is shown in terms of their KL-divergence (on logarithmic scale). Black line: KL-divergence between SI model and SD model. Red line: KL-divergence between MLLR-adapted model and SD model. Green line: KL-divergence between csMLLR-adapted model and SD model. The KL-divergence was approximated using Monte-Carlo sampling with 10^5 samples.

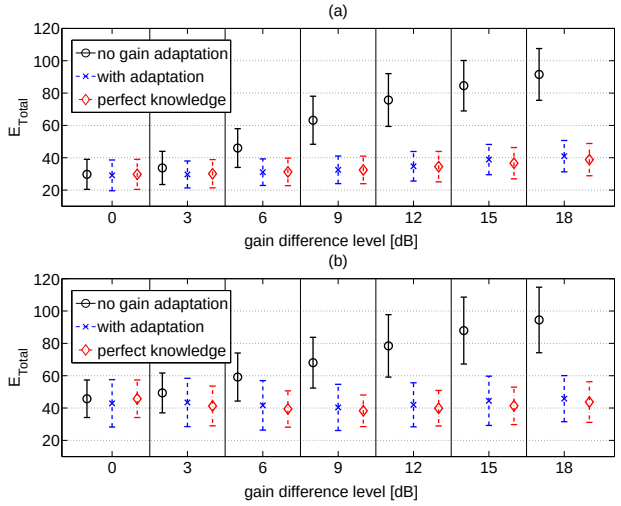

Fig. 7. Performance results for gain adaptation of three methods in terms of E_{Total}; 'no gain adaptation': Speaker models were used without adaptation. 'with adaptation': gain adaptation was performed prior to multi-pitch tracking using csMLLR. 'perfect knowledge': The speaker models were adapted by the true gain, i.e. the gain factor used to create the test mixture. (a) SD models. (b) SI models.

csMLLR converges to MLLR performance for increasing yet moderate values of C (around $C = 30$). In this case, csMLLR reaches almost the same performance as MLLR while significantly less parameters need to be learned (C^2+C instead of $D^2 + D$).

For the case of little adaptation data (i.e. $1.8\,$s) and low values of C, csMLLR is able to perform a meaningful transformation. This is shown in Figure 6, where the similarity between adapted speaker GMMs and SD GMMs is evaluated in terms of the KL-divergence for a range of relevant pitch states. In all cases, the csMLLR-adapted GMMs are more similar to the corresponding SD GMM than the SI GMM, i.e. the KL-divergence is smaller for these cases. In contrast, the KL-divergence of the MLLR-adapted GMMs is much larger, i.e. the GMMs are less similar to the corresponding SD GMM than the SI GMMs.

2) Gain Adaptation: In earlier experiments, we assumed that the gain level (i.e. energy) of training samples matches the testing conditions. In practice, however, this assumption does not hold – and any gain mismatch results in a degraded tracking performance. In this section, we evaluate the capability of the proposed adaptation framework to compensate gain mismatches between speaker models and the actual conditions in a test mixture composed of two simultaneous speakers. As described in Section IV-B1, gain adaptation can be performed by setting the spectral smoothness parameter of csMLLR to $C = 1$. Each test mixture was created with a predefined gain difference level Δ in the range of $\Delta = \{0, 3, 6, 9, 12, 15, 18\}$[dB]. For each value of Δ, and each combination of test utterances $s_1[n]$ and $s_2[n]$, a test mixture $y[n]$ was created according to $y[n] = 10^{\frac{\Delta}{20}} s_1[n] + s_2[n]$. For each gain level, we evaluated 135 test mixtures, as described in Section V-A. For each test mixture, we performed gain adaptation using the approximate csMLLR-based adaptation algorithm (see Section IV-B3). The pruning parameter was set to $R = 1000$, as this value provides a good tradeoff between accuracy and computational runtime (cf. results from Section V-C). The EM algorithm was run for 30 iterations. In many cases, no further increase of the lower bound was observed at less iteration numbers. Gain adaptation and subsequent multi-pitch tracking was performed either with SD or SI speaker models. The resulting performance in terms of E_{Total} is summarized in Figure 7. Without gain adaptation, the error increases significantly with rising gain difference level. With gain adaptation, we reach in all cases almost the same performance as with gain levels known a priori (i.e. perfect knowledge).

3) Self-Adaptation on Speech Mixtures Recorded in Reverberant Room: Similar as for gain adaptation, adaptation is performed on the same (short) test mixture for multi-pitch tracking. We use the proposed framework to perform self-adaptation on mixtures recorded in a real office environment. In such a scenario, the spectral characteristics of each source signal have changed due to multipath propagation or a different microphone transfer function. The mismatch between prior SD models obtained from close-talk microphone recordings and modified spectral characteristics in far-distance recordings results in a deteriorated multi-pitch tracking performance.

For this experiment, we used recordings where a set of test utterances from the GRID database was played through Yamaha MSP5A loudspeakers. The recording room has the dimensions $6.02 \times 5.32 \times 3\,$m. One of the walls of the room has a large window, and the floor is covered with a carpet. The measured reverberation time (RT_{60}) was $\mathrm{RT}_{60} \approx 500$ ms; no particular effort was made to reduce the reverberation. For each recorded speech mixture, two GRID utterances were played back simultaneously with two loudspeakers positioned at different locations around a circular microphone array (with $0.15\,$m diameter). We process the recordings of one channel of this array. The distance between loudspeakers and the microphone was about $2\,$m.

We used a total of 27 recorded test mixtures, consisting of 9 randomly chosen test mixtures from three speaker pairs each (female-female, male-female, male-male). For each test mixture, we applied self-adaptation using either csMLLR or

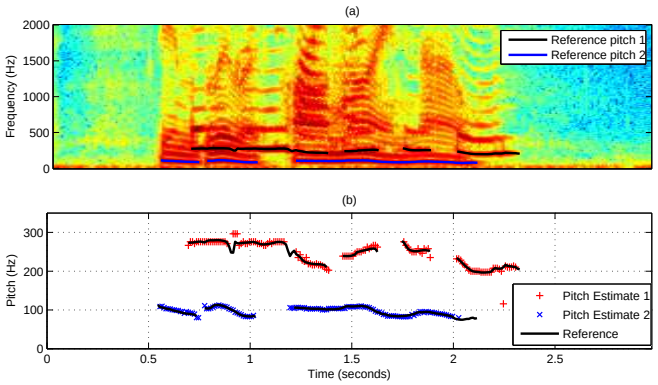

Fig. 8. Tracking result on synthetic mixture of one male and one female speaker. (a) Spectrogram of synthetic speech mixture, together with both reference pitch trajectories. (b) Estimated pitch trajectories using SD speaker models. The overall error E_{Total} is 12.2.

MLLR and evaluated the resulting multi-pitch tracking performance. A summary of the results is shown in Figure 10, where the performance is additionally compared to the case where (i) no adaptation is performed and (ii) multi-pitch tracking is performed on the equivalent synthetic test mixture.[14] SD models without adaptation work very well when applied to a synthetic mixture (see Figure 8), but result in heavily degraded performance when applied to the recorded mixture (see Figure 9). Self-adaptation is able to improve the performance, however the optimal choice of smoothing parameter C varies among speaker pairs. Generally, a low value of C works better, as only few data for adaptation is available. Self-adaptation clearly works best for mixtures of a male and female speaker, using csMLLR with $C = 3$. In Figure 8, the tracking result on a synthetic male-female mixture is shown. As there is no significant mismatch between SD models and the test condition, multi-pitch estimation works reasonably well. Tracking results with and without self-adaptation on the real recording of the same test mixture are shown in Figure 9. Finally, we note that the most general scenario where SI models are adapted on a speech mixture remains a difficult problem. One apparent problem with this scenario is the symmetry problem. When initializing each speaker model to identical SI models, the EM algorithm described in Section IV-B3 will produce identical transformation parameters for all models, i.e. it cannot converge to individual speaker models. One possible remedy is based on the identification of speech segments where only a single speaker is present. These segments can be used to pre-adapt an SI model to the specific (speaker) characteristics. However, this remains an issue for future research.

VI. Conclusions

We developed a model adaptation framework for FHMMs based on the EM algorithm and the MLLR technique to compensate any mismatch between training and testing conditions. We are able to adapt our models to novel environmental conditions, e.g. specific speakers, varying gain and/or acoustic environment. We propose a modification of the MLLR technique, where the adaptation of model parameters is constrained to modifications of the spectral envelope. This is beneficial in cases of few adaptation data. Furthermore, we develop a method for approximate inference based on likelihood pruning using a computationally efficient upper and lower bound on the observation likelihood. All developed methods are empirically compared for multipitch tracking. These FHMM models for handling single-channel multiple interacting sources are also appealing for related fields requiring signal separation. Examples are resolving interactions in brain-scan images or seismic data. Future work is concerned with the most general self-adaptation scenario, i.e. adapting SI models. Furthermore, we plan to apply the proposed framework on other applications such as speech recognition, source separation and speaker identification.

[14]For each recorded test mixture, a corresponding synthetic test mixture was created by linear superposition of the time-aligned original GRID utterances.

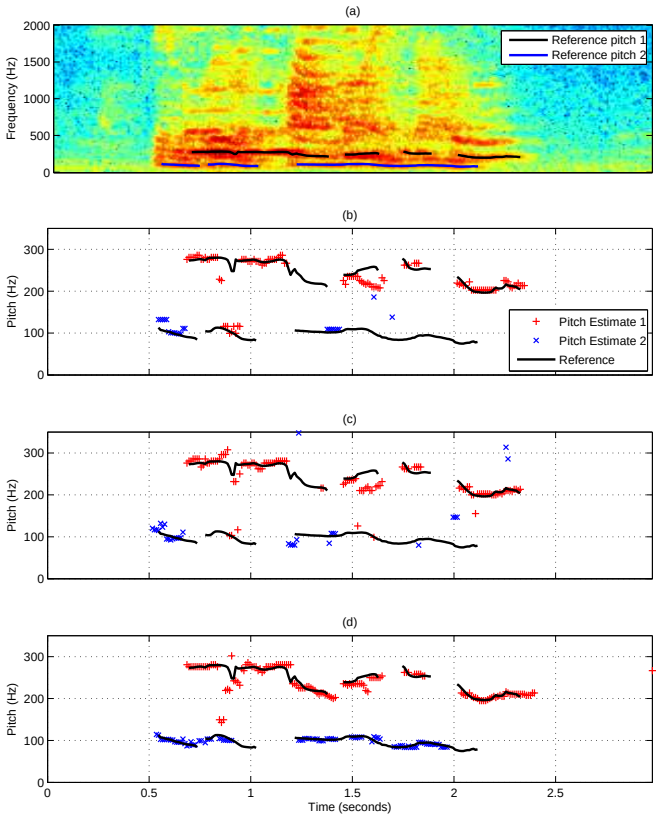

Fig. 9. Tracking results on real recording with and without self-adaptation. The same test utterances were used as for the synthetic mixture shown in Figure 8. (a) Spectrogram of recorded speech mixture. (b) Estimated pitch trajectories using SD speaker models without self-adaptation. (c) Estimated pitch trajectories after self-adaptation with MLLR. (d) Estimated pitch trajectories after self-adaptation with csMLLR ($C = 3$). The overall error E_{Total} achieved by the three methods is 65.2, 57.0 and 32.8, respectively.

Appendix

A. Derivation of the Upper Bound

Lemma 1: Let A_i and B_i be nonnegative real numbers and $i = 1, \ldots, K$, where $K \in \mathbb{N}$. Then

$$\prod_{i=1}^{K} (A_i + B_i) \geq \sum_{i=1}^{K} A_i \prod_{j \neq i} B_j. \tag{45}$$

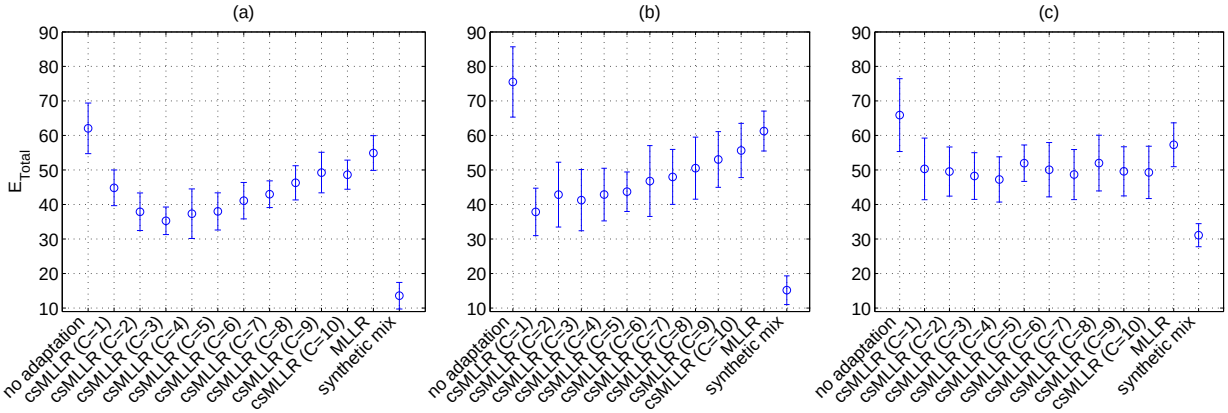

Fig. 10. Multi-pitch tracking performance in terms of E_{Total} after self-adaptation of SD models on real recordings. (a) male-female mixtures. (b) female-female mixtures. (c) male-male mixtures. Self-adaptation and subsequent multi-pitch tracking was performed for each test mixture separately. 9 test mixtures were used per speaker pair, and error bars indicate the corresponding mean and standard deviation of E_{Total} for various methods. 'no adaptation': SD models of both speakers were used without adaptation. 'csMLLR': SD models of both speakers were adapted on the test mixture using csMLLR. 'MLLR': SD models of both speakers were adapted on the test mixture using MLLR. 'synthetic mix': SD models of both speakers were used without adaptation and applied for multi-pitch tracking on the synthetic mixture (i.e. no recording in room environment).

Proof: By the multi-binomial theorem, which is a straightforward extension of the binomial theorem to the product of independent binomials, it holds that

$$\prod_{i=1}^{K}(A_i+B_i)=\sum_{n_1=0}^{1}\cdots\sum_{n_K=0}^{1}\prod_{i=1}^{K}A_i^{n_i}B_i^{1-n_i}, \tag{46}$$

where the right hand side consists of 2^K terms. Selecting only those K terms where exactly one index n_k is 1 and all other indices are zero, and exploiting the nonnegativity of A_i and B_i, we obtain the lower bound

$$\sum_{n_1=0}^{1}\cdots\sum_{n_K=0}^{1}\prod_{i=1}^{K}A_i^{n_i}B_i^{1-n_i}\geq\sum_{i=1}^{K}A_i\prod_{j\neq i}B_j. \tag{47}$$

■

Corollary 1: The likelihood $L(\cdot)$ obeys

$$\begin{aligned}L(\{x_k\},\{m_k\})&=\prod_{d=1}^{D}\sum_{k}\mathcal{N}_{k,x_k}^{m_k,d}\prod_{j\neq k}\Phi_{j,x_j}^{m_j,d}\\&\leq\prod_{d=1}^{D}\prod_{k}\left\{\mathcal{N}_{k,x_k}^{m_k,d}+\Phi_{k,x_k}^{m_k,d}\right\}.\end{aligned} \tag{48}$$

Proof: Follows directly from Lemma 1. ■

By applying the logarithm to (48), we finally obtain (8).

B. Derivation of the Lower Bound

We make use of the following two well known results (proofs can be found in [39]):

Lemma 2: For $i=1,\ldots,K$ with $K\in\mathbb{N}$, let $A_i\in\mathbb{R}$. Then

$$\ln\sum_{i=1}^{K}\exp A_i\geq\max_i A_i. \tag{49}$$

Lemma 3: Let $A_{i,j}\in\mathbb{R}$ and $i=1,\ldots,K$, $j=1,\ldots,N$ with $K\in\mathbb{N}$, $N\in\mathbb{N}$. Then

$$\max_i\left\{\sum_j A_{i,j}\right\}\leq\sum_j\max_i A_{i,j}. \tag{50}$$

We obtain (9) by rewriting the log-likelihood and applying Lemma 2 and 3:

$$\begin{aligned}\ln L(\{x_k\},\{m_k\})&=\ln\prod_{d=1}^{D}\sum_{k}\mathcal{N}_{k,x_k}^{m_k,d}\prod_{j\neq k}\Phi_{j,x_j}^{m_j,d}\\&=\sum_{d=1}^{D}\ln\left\{\left(\sum_k\frac{\mathcal{N}_{k,x_k}^{m_k,d}}{\Phi_{k,x_k}^{m_k,d}}\right)\prod_k\Phi_{k,x_k}^{m_k,d}\right\}\\&=\sum_d\left(\ln\sum_k\exp\left\{\ln\mathcal{N}_{k,x_k}^{m_k,d}-\ln\Phi_{k,x_k}^{m_k,d}\right\}+\sum_k\ln\Phi_{k,x_k}^{m_k,d}\right)\\&\geq\sum_d\left(\max_k\left\{\ln\mathcal{N}_{k,x_k}^{m_k,d}-\ln\Phi_{k,x_k}^{m_k,d}\right\}+\sum_k\ln\Phi_{k,x_k}^{m_k,d}\right)\\&=\sum_d\max_k\left\{\ln\mathcal{N}_{k,x_k}^{m_k,d}+\sum_{j\neq k}\ln\Phi_{j,x_j}^{m_j,d}\right\}\\&\geq\max_k\left\{\sum_d\ln\mathcal{N}_{k,x_k}^{m_k,d}+\sum_{j\neq k}\sum_d\ln\Phi_{j,x_j}^{m_j,d}\right\}.\end{aligned} \tag{51}$$

REFERENCES

[1] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *Journal of Acoustical Society of America*, vol. 25, pp. 975–979, 1953.

[2] J. Benesty, M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Springer-Verlag New York, Inc., 2007.

[3] P. Loizou, *Speech Enhancement: Theory and Practice*. CRC Press, 2007.

[4] D. Wang and G. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. John Wiley and Sons, 2006.

[5] S. T. Roweis, "Factorial models and refiltering for speech separation and denoising," in *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech)*, 2003, pp. 1009–1012.

[6] ——, "One microphone source separation," in *Advances in Neural Information Processing Systems (NIPS'00)*. MIT Press, 2000.

[7] M. Nakano, J. Le Roux, H. Kameoka, T. Nakamura, N. Ono, and S. Sagayama, "Bayesian nonparametric spectrogram modeling based on infinite factorial infinite hidden markov model," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*, oct. 2011, pp. 325 –328.

[8] G. Mysore, P. Smaragdis, and B. Raj, "Non-negative hidden markov modeling of audio with application to source separation," in *Latent Variable Analysis and Signal Separation*, ser. Lecture Notes in Computer Science, V. Vigneron, V. Zarzoso, E. Moreau, R. Gribonval, and E. Vincent, Eds. Springer Berlin Heidelberg, 2010, vol. 6365, pp. 140–148.

[9] A. Ozerov, C. Fevotte, and M. Charbit, "Factorial scaled hidden markov model for polyphonic audio representation and source separation," in *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09. IEEE Workshop on*, oct. 2009, pp. 121 –124.

[10] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[11] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Trans. Audio Speech and Language Processing*, vol. 15, no. 1, pp. 1–12, Jan 2007.

[12] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.

[13] M. Cooke, J. R. Hershey, and S. J. Rennie, "Monaural speech separation and recognition challenge," *Computer Speech and Language*, vol. 24, pp. 1–15, 2010.

[14] S. Rennie, J. Hershey, and P. Olsen, "Single-channel multitalker speech recognition," *IEEE Signal Processing Magazine*, vol. 27, no. 6, pp. 66 –80, 2010.

[15] J. Hershey, S. J. Rennie, P. Olsen, and T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Computer, Speech & Language*, vol. 24, pp. 45–66, 2010.

[16] M. Wohlmayr, M. Stark, and F. Pernkopf, "A probabilistic interaction model for multipitch tracking with factorial hidden Markov models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 799 –810, 2011.

[17] A. de Cheveigné, "Separation of concurrent harmonic sounds: Fundamental frequency estimation and a time-domain cancellation model of auditory processing," *The Journal of the Acoustical Society of America*, vol. 93, no. 6, pp. 3271–3290, 1993.

[18] G. Hu and D. Wang, "Monaural speech segregation based on pitch tracking and amplitude modulation," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1135 – 1150, 2004.

[19] M. Christensen and A. Jakobsson, *Multi-Pitch Estimation*, ser. Synthesis Lectures on Speech and Audio Processing, B.H.Juang, Ed. Morgan & Claypool, 2008.

[20] W. Hess, *Pitch determination of speech signals: Algorithms and devices*. Springer Verlag, 1983.

[21] Z. Ghaharamani and M. Jordan, "Factorial hidden Markov models," *Machine Learning*, vol. 29, pp. 245–275, 1997.

[22] A. Varga and R. Moore, "Hidden Markov model decomposition of speech and noise," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 845–848 vol.2, 1990.

[23] A. Nadas, D. Nahamoo, and M. Picheny, "Speech recognition using noise-adaptive prototypes," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 10, pp. 1495–1503, 1989.

[24] M. Wohlmayr, "Probabilistic model-based multiple pitch tracking of speech," Ph.D. dissertation, Graz University of Technology, 2012.

[25] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. B30, pp. 1–38, 1977.

[26] G. McLachlan and K. Basford, *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.

[27] S. Rennie, J. Hershey, and P. Olsen, "Variational loopy belief propagation for multi-talker speech recognition," in *International Conference on Spoken Language Processing (Interspeech)*, 2009, pp. 1331–1334.

[28] C. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 2, pp. 171 – 185, 1995.

[29] M. Gales and P. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech & Language*, vol. 10, no. 4, pp. 249 – 264, 1996.

[30] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994.

[31] R. Kuhn, J.-C. Junqua, P. Nguyen, and N. Niedzielski, "Rapid speaker adaptation in eigenvoice space," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 695 –707, 2000.

[32] R. Rose, E. Hofstetter, and D. Reynolds, "Integrated models of signal and background with application to speaker identification in noise," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 245–257, 1994.

[33] R. Weiss and D. Ellis, "A variational EM algorithm for learning eigenvoice parameters in mixed signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 113 –116.

[34] T. Virtanen, "Speech recognition using factorial hidden Markov models for separation in the feature space," in *International Conference on Spoken Language Processing (Interspeech)*, 2006, pp. 89–92.

[35] M. Radfar, W. Wong, R. Dansereau, and W.-Y. Chan, "Scaled factorial hidden Markov models: A new technique for compensating gain differences in model-based single channel speech separation," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 1918 –1921.

[36] S. Rennie, P. Olsen, J. Hershey, and T. Kristjansson, "The Iroquois model: Using temporal dynamics to separate speakers," in *Workshop on Statistical And Perceptual Audition*, 2006, pp. 24–30.

[37] M. Wohlmayr and F. Pernkopf, "EM-based gain adaptation for probabilistic multipitch tracking," in *International Conference on Spoken Language Processing (Interspeech)*, 2011, pp. 1969–1972.

[38] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[40] K. Petersen and M. Pedersen, *The Matrix Cookbook*, 2008.

[41] T. Moon and W. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, 2000.

[42] R. Neal and G. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in Graphical Models*, M. Jordan, Ed. Kluwer Academic Publisher, 1998, pp. 355–368.

[43] G. Pirker, M. Wohlmayr, S. Petrik, and F. Pernkopf, "A pitch tracking corpus with evaluation on multipitch tracking scenario," in *International Conference on Spoken Language Processing (Interspeech)*, 2011, pp. 1509–1512.

[44] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 120, pp. 2421–2424, 2005.

[45] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Kleijn W.B. and Paliwal K.K. [Ed], Speech Coding and Synthesis, Elsevier Science*, pp. 495–518, 1995.

[46] M. Wu, D. Wang, and G. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 229–241, 2003.

Michael Wohlmayr obtained his MSc degree from Graz University of Technology in 2007. He conducted his Master thesis in collaboration with University of Crete, Greece. In 2012 he received the PhD degree at the Signal Processing and Speech Communication Laboratory at Graz University of Technology, where he currently does his postdoctoral research. He has previously been with SYNVO, where he worked on speech synthesis. His research interests include Bayesian networks, speech and audio analysis, as well as statistical pattern recognition.

Franz Pernkopf received his MSc (Dipl. Ing.) degree in Electrical Engineering at Graz University of Technology, Austria, in summer 1999. He earned a PhD degree from the University of Leoben, Austria, in 2002. In 2002 he was awarded the Erwin Schrödinger Fellowship. He was a Research Associate in the Department of Electrical Engineering at the University of Washington, Seattle, from 2004 to 2006. Currently, he is Associate Professor at the Laboratory of Signal Processing and Speech Communication, Graz University of Technology, Austria. His research interests include machine learning, discriminative learning, graphical models, feature selection, finite mixture models, and image- and speech processing applications.