Analysis of Speaker Diarization Based on Bayesian HMM With Eigenvoice Priors

Mireia Diez, Lukáš Burget, *Member, IEEE*, Federico Landini, and Jan Černocký, *Senior Member, IEEE*

***Abstract*—In our previous work, we introduced our Bayesian Hidden Markov Model with eigenvoice priors, which has been recently recognized as the state-of-the-art model for Speaker Diarization. In this article we present a more complete analysis of the Diarization system. The inference of the model is fully described and derivations of all update formulas are provided for a complete understanding of the algorithm. An extensive analysis on the effect, sensitivity and interactions of all model parameters is provided, which might be used as a guide for their optimal setting. The newly introduced speaker regularization coefficient allows us to control the number of speakers inferred in an utterance. A naive speaker model merging strategy is also presented, which allows to drive the variational inference out of local optima. Experiments for the different diarization scenarios are presented on CALLHOME and DIHARD datasets.**

***Index Terms*—Speaker diarization, variational Bayes, hidden Markov models, clustering.**

I. Introduction

SPEAKER Diarization (SD) is the task of determining speaker turns in an audio recording of a conversation. In this paper, we present a Bayesian approach to SD, where the sequence of speech features representing a conversation is assumed to be generated from a Bayesian Hidden Markov Model (HMM). HMM states represent speakers and the transitions between the states correspond to speaker turns. The speaker (or HMM state) specific distributions are modeled by Gaussian Mixture Models (GMMs). In order to robustly learn the speaker-specific distributions, a strong informative prior is imposed on the GMM parameters, which makes use of eigenvoices just like i-vectors [1] or Joint Factor Analysis (JFA) [2] – the standard techniques for speaker recognition. Such prior facilitates discrimination between speaker voices in an input recording. The proposed Bayesian model offers a very elegant approach to SD as a straightforward and efficient Variational Bayes (VB) inference in a single probabilistic model addresses the complete SD problem. The system contrasts with most of the conventional approaches, where different models, techniques and heuristics are used to address the individual subproblems of SD such as speaker turn detection, speaker clustering or determining the number of speakers in the conversation.

We combine previously published ideas in our model. In particular, we use the eigenvoice prior for the speaker-specific GMMs as proposed in [3] and Bayesian HMM to model speaker turns similar to [4]. To point out the advantages (as well as possible limitations) of the proposed approach, we provide its comparison with the previous relevant SD approaches in the rest of this section. For a more complete overview of SD techniques, we kindly refer the reader to the excellent review papers [5], [6]. In the following text we assume that the reader has basic understanding of Variational inference in Bayesian models as presented for example in [7].

Most of the current practical SD systems address the task in the following steps: The parts of the input recording that are not of the interest are first removed (e.g. non-speech and overlapped speech). The conversation is then segmented into (preferably) speaker homogeneous segments. These segments are then clustered so that each cluster ideally contains all segments of exactly one speaker. While the early SD systems performed only these steps [8]–[11], more recent approaches use this initial clustering to train speaker-specific GMMs, which are then used to re-assign speech frames to speakers. For this purpose, an ergodic HMM is typically constructed with the speakers' GMMs as the state distributions and the speech frames are aligned to the states using the Viterbi alignment. Such re-segmentation was shown to greatly improve the diarization performance [12], [13]. Finally, clustering can be applied to the newly obtained speech segments and the two steps, clustering and re-segmentation, which generally involve two different models, can be iterated to benefit from each other's refinement. Although the models and techniques for the segmentation and clustering have evolved over time [5], [6] this general schema can be found in the majority of the state-of-the-art SD systems.

In contrast, the approach proposed in this paper does not use any dedicated step (or model) for the speaker clustering. We use a single HMM conceptually similar to the re-segmentation model (i.e. states represent speakers), which is, however, directly trained in an unsupervised way on the input recording. A random soft assignment of frames to HMM states can be used to initialize the training. Alternatively, another (possibly less accurate) SD system can be used for this initialization. Then, each VB training

Manuscript received March 23, 2019; revised August 26, 2019; accepted November 10, 2019. Date of publication November 22, 2019; date of current version December 24, 2019. This work was supported in part by European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Grant agreement 748097, in part by Czech National Science Foundation (GACR) project "NEUREM3" 19-26934X, in part by Czech Ministry of Interior project VI20152020025 "DRAPAK," in part by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602", and in part by research contract with Ericsson. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Lei Xie. *(Corresponding author: Mireia Diez.)*

The authors are with the Faculty of Information Technology, Brno University of Technology, 601 90 Brno, Czech Republic (e-mail: mireia@fit.vutbr.cz; burget@fit.vutbr.cz; landini@fit.vutbr.cz; cernocky@fit.vutbr.cz).

Digital Object Identifier 10.1109/TASLP.2019.2955293

iteration refines the HMM state-specific speaker models and re-calculates the soft (probabilistic) assignment of frames to HMM states. The informative eigenvoice prior forces HMM states to have only valid speaker distributions and the complexity control inherent in the Bayesian learning favors having exactly one such distribution per speaker. In other words, the model favors good correspondence between HMM states and speakers in the input conversation. Note that, for the conventional approaches, some heuristic needs to be used to stop the clustering process in order to determine the right number of speakers in the utterance. Our model allows us to determine the number of speakers in a principled way relaying again on the complexity control of the Bayesian learning. In particular, we use the principle of Automatic Relevance Determination (ARD) [7], which automatically learns to drop the redundant speaker distributions (HMM states) during the VB training. We only need to make sure that the training starts with a sufficient number of initial speaker distributions (i.e. an upper estimate of the number of speakers in the conversation).

Inspired by the success of i-vectors in speaker recognition [1], the SD system proposed in [14] used i-vectors as low-dimensional fixed-sized representations of speech segments in order to facilitate their clustering into speaker clusters. Since then, i-vectors have often been used in a similar manner for the SD task. For example, in [15] the input conversation is segmented into two-second long overlapping segments. For each segment, an i-vector is extracted and the i-vectors are clustered using the Agglomerative Hierarchical Clustering (AHC). For this clustering, Probabilistic Linear Discriminant Analysis (PLDA) [16] is used to measure the similarity between i-vectors, which is another standard technique borrowed from the speaker recognition field [17]. The underlying probabilistic model for i-vector extraction is essentially the same as the one used for modeling speaker-specific distributions in our SD system (i.e. GMM with eigenvoice prior, where the speaker-specific distribution is represented by an i-vector-like low-dimensional latent variable). However, the SD systems based on i-vector clustering have the following disadvantage compared to our approach: i-vectors cannot be extracted reliably as they are estimated on very short segments, which are not always speaker-homogeneous. Based on these suboptimal segment representations, hard decisions are made during the clustering process, which leads to errors that the AHC cannot recover from. In contrast, with our model, we have only a limited number of i-vector-like latent variables, each representing the distribution of one speaker. In each VB iteration, these latent variables are re-estimated on all the speech frames aligned with the corresponding speaker (or HMM state). The VB inference relies on soft probabilistic alignment of frames to speakers, which avoids making any hard decisions similar to AHC. To further avoid hard decisions, the VB inference takes into account the uncertainty of the latent variable estimates (it uses estimates of the posterior distributions of the latent variables). This is in contrast with i-vectors, which are Maximum a-posteriori (MAP) point estimates.

In latest works on SD, driven by the success of the DNN based techniques for SR [18], x-vector embeddings are being used in a similar fashion as the i-vectors in AHC systems [19]. These systems are very competitive but, unlike our approach, require extensive amounts of training data, which are not available for all kinds of scenarios. Besides, they have the same disadvantage described for the i-vector clustering approach. In fact, the best performing systems for both tracks in the last DIHARD challenge [20], used x-vector AHC based systems as an initialization for our VB-HMM based diarization system to achieve optimal results [21], [22].

A similar VB approach to SD was first proposed in [23], [24] and further extended by adding the eigenvoice prior in [3]. These works, however, still assume that the conversation is pre-segmented into many short segments. The Bayesian model is only used to cluster such segments without making the AHC-like hard decisions. To obtain good diarization performance, such clustering still needs to be followed by the HMM-based re-segmentation step. Note that in our approach, such HMM is an integral part of the Bayesian model, which allows addressing both problems – clustering and re-segmentation – simultaneously.

Sticky Hierarchical Dirichlet Process HMM (HDP-HMM) was proposed for SD in [4], which is a model similar to our Bayesian HMM. The authors of [4], however, did not use the eigenvoice priors. Unlike our Bayesian HMM, HDP-HMM is a non-parametrical Bayesian model, which does not impose any upper limit on the number of speakers in the conversation (i.e. it represents an HMM with a potentially infinite number of states). On the other hand, the inference in HDP-HMM is more difficult. In [4], block Gibbs sampling was used for the inference, which is much less efficient than the VB inference used for our model. Our Bayesian model with fixed number of HMM states (i.e. limiting the maximum number of speakers) does not introduce any practical restrictions if the number of states is set sufficiently large.

An open source Python implementation of our SD approach is available [25].[1] This paper extends our previous work [26], in which the Variational Bayes diarization system with HMM priors was described for the first time. The novel contributions of this paper are as follows: we provide additional derivations for the complete overview of the inference in the model. We also present a more complete analysis of the parameters of the model to provide potential users with a better insight into how to tune the system for the optimal performance. We introduce an extension to the model – a set of factors scaling individual terms in the VB objective – which allows us to regulate the inference depending on the expected number of speakers. Also, a simple initialization method is introduced to remove the dependency on the external diarization systems which are normally used for the initialization of the VB inference. We also propose a speaker merging schema which attempts to avoid convergence to a local optimum. Finally, we give details on the implementation and complexity of the algorithm. The novelties of the system are analyzed on CALLHOME [27] and DIHARD [28] datasets.

II. The Model

Our model assumes that the sequence of observed speech features (e.g. Mel Frequency Cepstral Coefficients, MFCCs)

[1][Online]. Available: http://speech.fit.vutbr.cz/software/vb-diarization-eigenvoice-and-hmm-priors

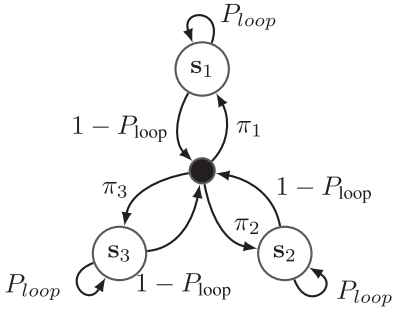

Fig. 1. HMM model for 3 speakers with 1 state per speaker, with a dummy non-emitting (initial) state.

corresponding to an input conversation is generated from an HMM with speaker-specific state distributions. The distribution of each speaker is modeled using a GMM with parameters constrained to live in an eigenvoice subspace (see Section II-B for more details). This allows us to robustly model the distribution of speaker s using only a low dimensional vector $\mathbf{y}_s$. We use an ergodic HMM with one-to-one correspondence between the HMM states and the speakers,[2] where transitions from any state to any state are possible. Note that our model does not consider any overlapped speech as each speech frame is assumed to be generated from an HMM state corresponding to only one of the S speakers. The transition probabilities are set in a way that discourages too frequent transitions between speakers in order to reflect speaker turn durations of a natural conversation. More details on setting and learning the transition probabilities can be found in Section II-A.

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ be the sequence of observed feature vectors and $\mathbf{Z} = \{z_1, z_2, \ldots, z_T\}$ the corresponding sequence of discrete latent variables defining the hard alignment of speech frames to HMM states. In our notation, $z_t = s$ indicates that the speaker (HMM state) s is responsible for generating observation $\mathbf{x}_t$ (note that this notation is not the same as in our previous work [26], where one-hot encoding was used for the latent variables $\mathbf{z}_t$).

To address the SD task using our model, the speaker distributions (i.e. the vectors $\mathbf{y}_s$) and the latent variables z_t are jointly estimated given an input sequence $\mathbf{X}$. The solution to the SD task is then given by the most likely sequence $\mathbf{Z}$, which encodes the alignment of speech frames to speakers.

A. HMM Topology

Given a particular initialization method (see Section V), we choose the number of states S higher than (or equal to) the maximum expected number of speakers. The HMM topology and transition probabilities model the speaker turn durations. The HMM model is ergodic (transitions between all states are possible). Fig. 1 shows an example of the HMM topology for only $S = 3$ speakers. The transition probabilities are set as follows: we transition back to the same speaker/state with probability P_{loop}. This probability is one of the tunable parameters in the model and will be typically set to a high value to discourage frequent speaker turns. The remaining probability $(1 - P_{\text{loop}})$ is the probability of changing speaker, which corresponds to the transition to the non-emitting node in Fig. 1. From the non-emitting node, we immediately transition to one of the speaker states with probability π_s.[3] Therefore, the probability of leaving a speaker and entering another speaker s is $(1 - P_{\text{loop}})\pi_s$. To summarize, the probability of transitioning from state s' to state s is

$$p(s|s') = (1 - P_{\text{loop}})\pi_s + \delta(s = s')P_{\text{loop}} \tag{1}$$

where $\delta(s = s')$ equals one if $s = s'$ and is 0 otherwise.

The non-emitting node in Fig. 1 is also the initial state of the model. Therefore, the probabilities π_s also control the selection of the initial HMM state (i.e. the state generating the first observation). These probabilities π_s are inferred (jointly with the variables $\mathbf{y}_s$ and z_t) from the input conversation. Thanks to the ARD principle [7] stemming from our Bayesian model, zero probabilities will be learned for the π_s corresponding to redundant speakers, which effectively drops such speakers from the HMM model. Typically, we initialize the HMM with a larger number of speakers and we make use of this behavior to drop the redundant speakers (i.e. to estimate the number of speakers in the conversation).

B. Speaker-Specific Distributions

For each speaker, the distribution of speech features is modelled using a GMM. Like similar models for speaker recognition (e.g. i-vector [1] or JFA [2]), our model assumes that the speaker-specific GMMs are all related to a single Universal Background Model (UBM-GMM). The UBM-GMM is an ordinary GMM typically trained on large amount of speech data from many speakers. All speaker-specific GMMs have the same number of Gaussian components C as the UBM-GMM. Furthermore, there is a one-to-one correspondence between the components of the UBM-GMM and the components of each speaker model. All speaker-specific GMMs share the same component weights w_c^{ubm} and covariance matrices $\mathbf{\Sigma}_c^{\text{ubm}}$ with the corresponding UBM-GMM components $c = 1..C$. Only the component mean vectors $\boldsymbol{\mu}_{sc}$ take speaker-specific values, which are however still constrained as follows: Let $\boldsymbol{\mu}_s = [\boldsymbol{\mu}_{s1}^T \ \boldsymbol{\mu}_{s2}^T \ldots \boldsymbol{\mu}_{sC}^T]^T$ be the super-vector of concatenated Gaussian component means for speaker s and let $\boldsymbol{\mu}^{\text{ubm}}$ be the similarly defined super-vector of concatenated UBM-GMM means. The high-dimensional super-vectors

$$\boldsymbol{\mu}_s = \boldsymbol{\mu}^{\text{ubm}} + \mathbf{V}\mathbf{y}_s \tag{2}$$

are constrained to live in a low-dimensional subspace around the origin given by $\boldsymbol{\mu}^{\text{ubm}}$. The subspace is spanned by the so-called eigenvoice basis, columns of the low-rank matrix $\mathbf{V}$. This matrix

[2]In our previous work [26], a more general HMM topology with (possibly) multiple states per speaker was proposed to impose minimum speaker turn duration constraint. In this work, we only consider the special case of this topology with only one state per speaker as such configuration was found sufficient for obtaining the best performance.

[3]For convenience, we allow to re-enter the same speaker as it leads to simpler update formulas.

is also shared by all speaker models. The only speaker-specific parameters are then the low-dimensional vectors $\mathbf{y}_s$, which can be seen as coordinates of $\boldsymbol{\mu}_s$ in the low-dimensional subspace. All the speaker independent parameters $\boldsymbol{\mu}^{\text{ubm}}$, $\boldsymbol{\Sigma}_c^{\text{ubm}}$, w_c^{ubm} and $\mathbf{V}$ are pre-trained and fixed during the inference in our model when addressing the SD task. Therefore, the speaker-specific distributions

$$p(\mathbf{x}_t|\mathbf{y}_s) = \text{GMM}(\mathbf{x}_t; \{\boldsymbol{\mu}_{sc}\}, \{\boldsymbol{\Sigma}_c^{\text{ubm}}\}, \{w_c^{\text{ubm}}\}) \quad (3)$$

can be expressed only in terms of the low-dimensional vectors $\mathbf{y}_s$, which can be robustly estimated from the limited amount of speech available in the input conversation.

To further improve the robustness of the speaker model estimates, we treat $\mathbf{y}_s$ as a latent variable with standard normal prior

$$p(\mathbf{y}_s) = \mathcal{N}(\mathbf{y}_s; \mathbf{0}, \mathbf{I}). \quad (4)$$

Inserting such prior into (2) translates to a Gaussian prior imposed on speaker mean super-vectors

$$p(\boldsymbol{\mu}_s) = \mathcal{N}(\boldsymbol{\mu}_s; \boldsymbol{\mu}^{\text{ubm}}, \mathbf{V}\mathbf{V}^T), \quad (5)$$

which can also be seen as an informative prior on the possible speaker GMMs. To obtain such prior that correctly models the variability of the speaker mean super-vectors, the matrix $\mathbf{V}$ needs to be pre-trained on speech data from a large number of speakers. Note, that the model for representing speaker-specific distributions described above is essentially the same as the model for i-vector extraction [1] or JFA [2]. Therefore, we do not provide a detailed description of the procedure for training $\mathbf{V}$ in this paper and we kindly refer the reader to the original sources. In our experiments, we train $\mathbf{V}$ using exactly the same procedure (Expectation Maximization algorithm) and the same code that we normally use for training the total variability matrix for i-vector extraction in the speaker recognition task.

C. Bayesian HMM

To summarize, our complete model for SD is a Bayesian HMM, which is defined in terms of the state-specific distributions (or so-called output probabilities)

$$p(\mathbf{x}_t|z_t = s) = p(\mathbf{x}_t|s) = p(\mathbf{x}_t|\mathbf{y}_s) \quad (6)$$

described in Section II-B and the transition probabilities

$$p(z_t = s|z_{t-1} = s') = p(s|s') \quad (7)$$

described in Section II-A. By abuse of notation, $p(z_1|z_0)$ will correspond to the initial state probability $p(z_1 = s) = \pi_s$ in the following formulas.

The complete model can be also defined in terms of the joint probability of the observed and latent random variables (and their factorization) as

$$\begin{aligned} p(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) &= p(\mathbf{X}|\mathbf{Z}, \mathbf{Y})p(\mathbf{Z})p(\mathbf{Y}) \\ &= \prod_t p\left(\mathbf{x}_t|z_t\right) \prod_t p\left(z_t|z_{t-1}\right) \prod_s p\left(\mathbf{y}_s\right), \end{aligned} \quad (8)$$

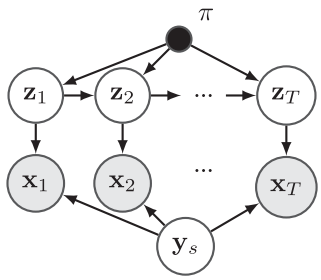

Fig. 2. Bayesian Network corresponding to our diarization model.

where $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_S\}$ is the set of all the speaker-specific latent variables. The corresponding Bayesian Network is depicted in Fig. 2.

The model assumes that each feature sequence corresponding to an input conversation is obtained using the following generative process:

for $\mathbf{s} = \mathbf{1}..\mathbf{S}$ **do**
  $\mathbf{y}_s \sim \mathcal{N}(0, \mathbf{I})$
  $\boldsymbol{\mu}_s = \boldsymbol{\mu}^{\text{ubm}} + \mathbf{V}\mathbf{y}_s$
for $t = 1..T$ **do**
  $z_t \sim p(z_t|z_{t-1})$
  $\mathbf{x}_t \sim p(\mathbf{x}_t|z_t)$

Here, first a speaker-specific GMM distribution is sampled for each speaker s. This is achieved by sampling the low dimensional speaker vectors $\mathbf{y}_s$ from the standard normal prior and then applying (2) to obtain the corresponding GMM means. Recall that the other GMM parameters are pre-trained and shared by all speaker models. Once the speaker models have been generated for the conversation, the initial HMM state is selected according to the distribution $p(z_1) = p(z_1|z_0)$. Given the selected state z_1, the first observation $\mathbf{x}_1$ is sampled from the distribution $p(\mathbf{x}_1|z_1)$ (i.e. the speaker-specific GMM corresponding to the state z_1). Then, for each frame t, a new HMM state is selected according to $p(z_t|z_{t-1})$ and a new observation $\mathbf{x}_t$ is sampled from $p(\mathbf{x}_t|z_t)$.

We call our model "Bayesian" HMM as we impose a prior on the parameters of the state distributions (i.e. $\mathbf{y}_s$ is a latent variable with standard normal prior). However, unlike other "Fully Bayesian" HMM implementations [4], [29], we do not impose any prior on the transition probabilities.

Further note that, although our state distributions are GMMs, the Bayesian model does not introduce any latent variables defining the alignments of observations to the Gaussian components. We assume that this alignment is exactly the same for all the speaker-specific GMMs and UBM-GMM. This is possible thanks to the correspondence between the Gaussian components in these models. Therefore, we pre-calculate the alignments using the UBM-GMM and consider them observed during the inference in our model. More precisely, we calculate soft alignments (or responsibilities) as the posterior probabilities of UBM-GMM components given observations $p_{\text{ubm}}(c|\mathbf{x}_t)$. See Appendix A for more details on this approximation. Note that

such approximation, which considerably simplifies the inference in the model, is also used in similar models for speaker recognition [1], [2].

III. Diarization Inference

A. Variational Bayesian Inference

The diarization problem consists in finding the assignment of frames to speakers, which is represented by the latent sequence $\mathbf{Z}$. In order to find the most likely sequence $\mathbf{Z}$, we need to infer the posterior distribution $p(\mathbf{Z}|\mathbf{X}) = \int p(\mathbf{Z}, \mathbf{Y}|\mathbf{X}) d\mathbf{Y}$. Unfortunately, the evaluation of this integral is intractable, and therefore, we will approximate it using Variational Bayes inference [7], where the distribution $p(\mathbf{Z}, \mathbf{Y}|\mathbf{X})$ is approximated by $q(\mathbf{Z}, \mathbf{Y})$. We use the mean-field approximation [3], [7] assuming that the approximate posterior distribution factorizes as

$$q(\mathbf{Z}, \mathbf{Y}) = q(\mathbf{Z})q(\mathbf{Y}). \tag{9}$$

The particular form of the approximate distributions $q(\mathbf{Z})$ and $q(\mathbf{Y})$ directly follows from the optimization described below.

We search for such $q(\mathbf{Z}, \mathbf{Y})$ that minimizes the Kullback-Leibler divergence $D_{KL}(q(\mathbf{Z}, \mathbf{Y}) \| p(\mathbf{Z}, \mathbf{Y}|\mathbf{X}))$, which is equivalent to maximizing the standard VB objective – the Evidence Lower Bound Objective (ELBO) [7]

$$\mathcal{L}\left(q(\mathbf{X}, \mathbf{Y})\right) = E_{q(\mathbf{Y},\mathbf{Z})}\left\{\ln\left(\frac{p(\mathbf{X}, \mathbf{Y}, \mathbf{Z})}{q(\mathbf{Y}, \mathbf{Z})}\right)\right\}. \tag{10}$$

Using the factorization (9), the ELBO can be split into three terms

$$\begin{aligned}\hat{\mathcal{L}}\left(q(\mathbf{X}, \mathbf{Y})\right) = {} & F_A E_{q(\mathbf{Y},\mathbf{Z})}\left[\ln p(\mathbf{X}|\mathbf{Y}, \mathbf{Z})\right] \\ & + F_B E_{q(\mathbf{Y})}\left[\ln \frac{p(\mathbf{Y})}{q(\mathbf{Y})}\right] + E_{q(\mathbf{Z})}\left[\ln \frac{p(\mathbf{Z})}{q(\mathbf{Z})}\right],\end{aligned} \tag{11}$$

where the first term is the expected likelihood of the observed feature sequence $\mathbf{X}$ and the second and third terms are Kullback-Leibler divergences $D_{KL}(q(\mathbf{Y}) \| p(\mathbf{Y}))$ and $D_{KL}(q(\mathbf{Z}) \| p(\mathbf{Z}))$ regularizing the approximate posterior distributions $q(\mathbf{Y})$ and $q(\mathbf{Z})$ towards the priors $p(\mathbf{Y})$ and $p(\mathbf{Z})$. In (11), we modified the ELBO by scaling the first two terms by constant factors F_A and F_B.[4] The theoretically correct values for these factors leading to the original ELBO (11) are $F_A = F_B = 1$. However, as explained in Section IV, choosing different values gives us finer control over the inference, which can be used to improve diarization performance.

B. Sufficient Statistics

As pointed out in the last section, our inference assumes that the alignmet of observations to GMM components is the same for all the speaker-specific GMMs and is defined in terms of UBM-GMM responsibilities $p_{\mathrm{ubm}}(c|\mathbf{x}_t)$. Given such assumption, the likelihood of an observed feature vector for a speaker-specific GMM $p(\mathbf{x}_t|\mathbf{y}_s)$ can be efficiently evaluated (see Appendix A) using the following per-frame zero, first and second order sufficient statistic:

$$\zeta_{tc} = p_{\mathrm{ubm}}(c|\mathbf{x}_t) \tag{12}$$

$$\boldsymbol{\rho}_t = \sum_c \zeta_{tc} \mathbf{V}_c^T \boldsymbol{\Sigma}_c^{\mathrm{ubm}\,-1} \left(\mathbf{x}_t - \boldsymbol{\mu}_c^{\mathrm{ubm}}\right) \tag{13}$$

$$\boldsymbol{\Phi}_t = \sum_c \zeta_{tc} \mathbf{V}_c^T \boldsymbol{\Sigma}_c^{\mathrm{ubm}\,-1} \mathbf{V}_c, \tag{14}$$

where $\mathbf{V}_c$ is the block of matrix $\mathbf{V}$ corresponding to the GMM component c. Hence, the same statistics also appear in the following update formulas, which are derived from $p(\mathbf{x}_t|\mathbf{y}_s)$.

C. Update Formulas

As described above, we search for the approximate posterior $q(\mathbf{Z}, \mathbf{Y})$ that maximizes the ELBO (11). In the case of the mean-field factorization (9), we proceed iteratively by finding the $q(\mathbf{Y})$ that maximizes the ELBO given fixed $q(\mathbf{Z})$ and vice versa. This section provides all the formulas necessary for implementing these updates or for understanding our open source Python implementation [25]. In this section, we do not give any details on deriving the update formulas. These derivations can be, however, found in Appendix B.

1) Updating $q(\mathbf{Y})$: Given a fixed $q(\mathbf{Z})$, the distribution over $\mathbf{Y}$ that maximizes the ELBO is (see Appendix B for derivation)

$$q^*(\mathbf{Y}) = \prod_s q^*(\mathbf{y}_s), \tag{15}$$

where the speaker-specific approximate posteriors

$$q^*(\mathbf{y}_s) = \mathcal{N}\left(\mathbf{y}_s|\boldsymbol{\alpha}_s, \mathbf{L}_s^{-1}\right) \tag{16}$$

are Gaussians with the mean vector and precision matrix

$$\boldsymbol{\alpha}_s = \frac{F_A}{F_B} \mathbf{L}_s^{-1} \sum_t \gamma_{ts} \boldsymbol{\rho}_t \tag{17}$$

$$\mathbf{L}_s = \mathbf{I} + \frac{F_A}{F_B} \sum_t \gamma_{ts} \boldsymbol{\Phi}_t. \tag{18}$$

In this update formula, $\gamma_{ts} = q(z_t = s)$ is the marginal approximate posterior derived from the current estimate of the distribution $q(\mathbf{Z})$ (see below), which can be interpreted as the responsibility of speaker s for generating observation $\mathbf{x}_t$ (i.e. defines a soft alignment of speech frames to speakers). Note that (17) and (18) correspond to the standard formulas for i-vector extraction [1], except for the responsibility term γ_{ts}, which would be always 1 for i-vectors as the standard speaker verification task assumes that all the frames in a recording come from a single speaker. Furthermore, i-vectors are only MAP point estimates of the latent variable (i.e. the means $\boldsymbol{\alpha}_s$), whereas our inference considers the whole posterior distributions (including the precisions $\mathbf{L}_s$) with the aim of accounting for the uncertainty in the speaker model estimates.

2) Updating $q(\mathbf{Z})$: We never need to infer the complete distribution over all the possible alignments of observations to speaker $q(\mathbf{Z})$. When updating $q(\mathbf{Y})$ using (17) and (18), we only need the marginals $\gamma_{ts} = q(z_t = s)$. Therefore, when updating $q(\mathbf{Z})$, we can directly search for the responsibilities γ_{ts}

[4]Note that similar scaling factor for the third term would be redundant as only relative scale of the three factors is relevant for the optimization.

that correspond to the distribution $q^*(\mathbf{Z})$ maximizing the ELBO given a fixed $q(\mathbf{Y})$ (see Appendix B for the complete derivation of $q^*(\mathbf{Z})$). Similar to the standard HMM training, such responsibilities can be calculated efficiently using a forward-backward algorithm as

$$\gamma_{ts} = \frac{A(t,s)B(t,s)}{\overline{p}(\mathbf{X})} \tag{19}$$

where the so-called forward probability

$$A(t,s) = \bar{p}(\mathbf{x}_t|s) \sum_{s'} A(t-1,s')p(s|s') \tag{20}$$

is recursively evaluated by progressing forward in time for t=1..T starting with $A(0,s) = \pi_s$. Similarly,

$$B(t,s) = \sum_{s'} B(t+1,s')\bar{p}(\mathbf{x}_{t+1}|s')p(s'|s) \tag{21}$$

is the backward probability evaluated using backward recursion for times $t = T..1$ starting with $B(T,s) = 1$.

$$\overline{p}(\mathbf{X}) = \sum_s A(T,s) \tag{22}$$

is the total forward probability and the term

$$\begin{aligned}
&\overline{p}(\mathbf{x}_t|s) \\
&= \exp\left\{ F_A \left[\boldsymbol{\alpha}_s^T \boldsymbol{\rho}_t - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Phi}_t \left[\mathbf{L}_s^{-1} + \boldsymbol{\alpha}_s\boldsymbol{\alpha}_s^T\right]\right) \right.\right. \\
&\quad - \frac{D}{2}\ln 2\pi - \sum_c \frac{\zeta_{tc}}{2}\ln|\boldsymbol{\Sigma}_c^{\mathrm{ubm}}| + \sum_c \zeta_{tc}\ln\frac{w_c^{\mathrm{ubm}}}{\zeta_{tc}} \\
&\quad \left.\left. -\frac{1}{2}\sum_c \zeta_{tc}\left(\mathbf{x}_t - \boldsymbol{\mu}_c^{\mathrm{ubm}}\right)^T \boldsymbol{\Sigma}_c^{\mathrm{ubm}^{-1}}\left(\mathbf{x}_t - \boldsymbol{\mu}_c^{\mathrm{ubm}}\right) \right] \right\}
\end{aligned} \tag{23}$$

is derived in (31) as the expected likelihood of observation $\mathbf{x}_t$ given a speaker s taking into account its uncertainty $q(\mathbf{y}_s)$.

3) Updating π_s: Finally, the speaker priors π_s are updated as Maximum Likelihood type II estimates [7]: Given fixed $q(\mathbf{Y})$ and $q(\mathbf{Z})$, we search for the values of π_s that maximize the ELBO (11), which gives the following update formula

$$\pi_s \propto \gamma_{1s} + \frac{(1-P_{\mathrm{loop}})\pi_s}{\overline{p}(\mathbf{X})} \sum_{t=2}^{T}\sum_{s'} A(t-1,s')p(\mathbf{x}_t|s)B(t,s) \tag{24}$$

with the constraint $\sum_s \pi_s = 1$. As described in Section II-A, this update tends to drive the π_s corresponding to "redundant speakers" to zero values, which effectively drops them from the model and selects the right number of speakers in the input conversation.

4) Evaluating the ELBO: The convergence of the iterative VB inference can be monitored by evaluating the ELBO objective. For the Bayesian HMM, the ELBO can be efficiently evaluated (see page 95 of [29]) as

$$\hat{\mathcal{L}} = \ln\overline{p}(\mathbf{X}) + \sum_s \frac{F_B}{2}\left(R + \ln|\mathbf{L}_\mathbf{s}^{-\mathbf{1}}| - tr(\mathbf{L}_s^{-1}) - \boldsymbol{\alpha}_s^T\boldsymbol{\alpha}_s\right) \tag{25}$$

where R is the rank of the eigenvoice matrix $\mathbf{V}$. This way of evaluating the ELBO is very practical as the term $\overline{p}(\mathbf{X})$ from (22) is obtained as a byproduct of "updating $q(\mathbf{Z})$" using the forward-backward algorithm. On the other hand, (25) allows to evaluate the ELBO only right after the $q(\mathbf{Z})$ update. It does not allow to monitor the improvement in ELBO obtained form $q(\mathbf{Y})$ or π_s updates, which might be useful for debugging purposes. Therefore, we also provide the derivation formulas for the explicit evaluation of all three ELBO terms from (11) in Appendix C.

The complete VB inference consisting of iterative updates of $q(\mathbf{Y})$, $q(\mathbf{Z})$ and parameters π_s is sumarized in the following algorithm:

Initialize all γ_{ts} as described Section V.
repeat
  Update $q(\mathbf{y}_s)$ for $s = 1..S$ using(16)
  for $t = 1..T$ **do**
    Calculate $A(t,s)$ for $s = 1..S$ using (20)
  for $t = T..1$ **do**
    Calculate $B(t,s)$ for $s = 1..S$ using (21)
  Update γ_{ts} for $t = 1..T, s = 1..S$ using(19)
  Update π_s for $s = 1..S$ using (24)
  Evaluate ELBO $\hat{\mathcal{L}}$ using (25)
until convergence of $\hat{\mathcal{L}}$

IV. Parameters of the Diarization Algorithm

In this section we provide a summary of all the parameters used to control the inference in our model, namely: P_{loop}, *downsamplingFactor*, F_A and F_B.

1) P_{loop} was introduced in Section II-A describing transition probabilities in our HMM. It is defined as the probability of looping in the same speaker state. It is typically set to high values (close to one) to discourage frequent speaker changes.

2) The *downsamplingFactor* was introduced to speed up the diarization algorithm. Formally, we assume a modified HMM generative process where *downsamplingFactor* observations are generated at once from the current HMM state in each step (i.e. after each transition). To reflect this model modification in the VB inference, we simply accumulate the per-frame statistics (12)–(14) for each *downsamplingFactor* consecutive frames, which effectively reduces the frame rate of the statistics by this factor. This modification can significantly speed up the VB inference for the price of reduced frame resolution leading to a coarser granularity of the output labeling. However, the reduced frame rate does not necessarily have to be seen as a disadvantage. In fact, it can help to improve modeling of speaker turn durations. With a single HMM state per speaker, the HMM assumes geometrically distributed speaker turn durations. In the case of 10 ms frame rate, as used for our MFCC features, such duration model does not reflect the reality very well. As pointed out in [4], however, for reduced frame rates (e.g. 250 ms corresponding to *downsamplingFactor* $= 25$), the geometric distribution becomes quite a good match for the speaker turn duration modeling.

3) The *Acoustic scaling factor* F_A is introduced to counteract the assumption of statistical independence between observations. This incorrect assumption made by the HMM results in overconfident posterior distributions of the latent variables. To counteract this problem in our previous work [26], we introduced a factor called $statScale$, which was used to scale the sufficient statistics from (12)–(14). This factor was typically set to a value between 0 and 1 in order to effectively reduce the number of observations, which makes the model believe that there is less evidence in the data for estimating the posterior distributions. In this work, we have introduced the factor F_A to weight the first term of the ELBO in (11) (i.e. the expected likelihood of the data), which has the same effect in the update formulas as the $stateScale$ factor from our previous work. Presenting F_A as a scaling factor in the ELBO allows us to introduce this factor directly into the update formulas in a proper formal way.

4) The *speaker regularization coefficient* F_B weights the second term of the ELBO in (11), the Kullback-Leibler divergence between the approximate speaker posterior and the speaker prior $D_{KL}(q(\mathbf{Y})\|p(\mathbf{Y}))$. This term can be seen as a regularization term penalizing the complexity of the speaker models (i.e. the posteriors of speaker latent variables should not be too far from the standard normal priors). Note that, when dropping speaker s from the model (by setting $\pi_s = 0$ and forbidding to enter the speaker's state), the speaker s has zero contribution to this D_{KL} term (with no observations aligned to speaker s, $q(\mathbf{y}_s) = p(\mathbf{y}_s)$). As a consequence, setting high value of F_B results in the VB inference dropping more speakers.

These parameters cannot be optimized individually, as they are interdependent in terms of the influence on the diarization error rate e.g. both P_{loop} and *downsamplingFactor* have an effect on speaker turn duration modeling. Therefore, these parameters have to be jointly optimized. This will be studied in Section VIII.

V. System Initializations

In our experiments, we start the VB inference by initializing the responsibilities γ_{ts}. This can be done either randomly, heuristically, or from the output labeling of an external diarization system. The initialization is provided in the form of a matrix of frame-wise speaker responsibilities γ_{ts}. If the input labeling is not probabilistic (as in any of the methods described below), it is transformed into this matrix by giving the selected speaker only a slightly higher probability than to the rest of the speakers.

The VB inference benefits from a good initialization, which seems to drive the inference into better solutions and help avoiding local optima. In previous works, we have used several techniques for the initialization. Namely, random initializations, i-vector PLDA AHC [26], [30] and x-vector PLDA AHC [22]. For the random initialization, a (maximum) number of speakers is chosen for the input conversation and a random assignment of frames to speakers is made. As shown in [26], this initialization method does not provide optimal results. However, comparable results to (or better than) those attained using external system-based initialization methods can be obtained if this random initialization is repeated several times (5 times was found to be sufficient) and the solution with the best ELBO value is selected. This strategy works for the following reason: each random initialization can lead to a different result, which corresponds to a different local optima. Re-starting the algorithm multiple times with the random initialization and selecting the solution with the best ELBO gives us the chance to select a good solution possibly close to the global optima. The effectiveness of this strategy indicates that the ELBO is a good indicator of the diarization performance.

In this paper, we use an alternative initialization that also does not require external diarization systems. We consider a simple method, which segments the input utterance into 5 second segments and assigns a different speaker label to each segment. We will refer to this initialization as *chunking*. Note that this initialization can result in slower VB inference, as the computational complexity of the inference scales linearly with the number of modeled speakers. We explore this method because of its simplicity, as it does not require extra logic to check the quality of the convergence of the algorithm.

VI. Speaker Model Merging

Although the VB inference has the ability of dropping redundant speakers, the algorithm might end up in a local optimum where speech of a single speaker is attributed to multiple HMM states (speaker models). In order to escape from such local optimum, we may try to merge a pair of speaker models and observe whether the ELBO improves. More precisely, we first let the VB inference converge, and then we iteratively consider all possible pairs of remaining speakers. For each pair, we merge the speaker models by summing the corresponding responsibilities γ_{ts}, we update the speaker model using (16) and re-estimate the speaker responsibilities for all speakers using (19). Then the Lower Bound objective (25) is re-evaluated. If the ELBO increases, we keep the merged diarization output (otherwise we revert to the previous solution). The process is repeated until no speaker model merging improves the ELBO. It is fast to merge a given pair of speakers and re-evaluating the model for such case (much faster than re-running the algorithm for another initialization). However, for each step of the above algorithm, we need to consider merging every possible pair of speakers, which may be very costly depending on the number of speakers that the VB diarization system converges to. Still, the approach provides a simple way of escaping from local optima, which improves results as will be shown in our experiments.

VII. Experimental Setup

A. CALLHOME Dataset and System Description

Our initial experiments are evaluated on the NIST SRE 2000 CALLHOME dataset [31], consisting of 500 recordings of conversational telephone speech. The number of speakers per recording ranges between 2 and 7, although 87% of the files contain only 2 or 3 speakers. It amounts to around 15 hours of speech (after Voice Activity Detection).

The features used in our experiments are the standard 19 MFCCs plus energy, with no deltas, extracted from 8 kHz speech. Neither mean nor variance normalization are applied in

TABLE I
DATA SOURCES IN DIHARD DATASET, DESCRIPTION AND AVERAGE NUMBER OF SPEAKERS PER RECORDING IN EACH DOMAIN

Set	Source	Description	spk per rec. Dev/Eval
Dev and Eval	ADOS	Clinical interviews (Children)	2.1 / 2.1
	DCIEM	Map Tasks	2 / 2
	LIBRIVOX	Audio books	1 / 1
	SCOTUS	Supreme Court oral arguments	6.9 / 7.3
	SEEDLINGS	Child language acquisition	2.9 / 3.6
	YP	Radio interviews	3.8 / 3.7
	VAST	YouTube videos	3.8 / 3.5
Only Dev	RT04S	Meeting speech	5.4
	SLX	Sociolinguistic interviews	3.5
Only Eval	ROAR	Meeting speech	3.9
	MIXER6	Sociolinguistic interviews	2
	CIR	Restaurant conversations	6.4

the feature extraction. The system employs gender-independent UBM-GMM with 1024 diagonal-covariance Gaussian components. The dimensionality of the speaker latent variable $\mathbf{y}_s$ is 400. To train the UBM-GMMs and i-vector extractors (and PLDA model used for initialization), we use the NIST SRE 2004–2008 datasets as in [15].

The CALLHOME dataset will be used to compare several VB inference initialization methods described in Section V. In case of initialization from the external i-vector/PLDA-AHC system, we followed the configuration employed in [26]: 64 dimensional i-vectors are projected by PCA (estimated on per-recording basis) to 3 dimensions and clustered using AHC with calibrated PLDA similarity score [17], [32] as the metric.

B. DIHARD Dataset and System Description

Further experiments are evaluated on the DIHARD I corpus developed for the first DIHARD challenge [20], designed to foster research on diarization in hard conditions including data from several sources (radio, YouTube, child language acquisition, etc.) [28]. The corpus consists of 164 development and 172 evaluation recordings, containing around 14 h and 17 h of speech, respectively. For a better understanding of the results, Table I introduces a summary and a brief description of the different sources present in the dataset.

In order to have comparable results, we keep the experimental setup similar to the one found optimal for the dataset in our previous work [22]. In this section, we give a brief summary of the setup. For more details, we refer the reader to the full system description in [22].

The Weighted Prediction Error (WPE) [33] method was used to remove late reverberation from the audio signal. The features used in our experiments are standard 19 MFCCs plus energy, plus delta, extracted from 16 kHz speech. Neither mean nor variance normalization are applied in the feature extraction. We use a gender-independent UBM-GMM, with 1024 diagonal-covariance Gaussian components. The dimensionality of the speaker latent variable $\mathbf{y}_s$ is 400.

We consider two sets for training the UBM-GMMs and the i-vector extractor: 1) As in the case of our submission to the DIHARD challenge [22], we use a reduced DIHARD development set (14 h of speech) excluding audios coming from YouTube (VAST) plus the evaluation set (17 h of speech) to train the UBM, and only the reduced development set to train the i-vector extractor [22], [28]. 2) Inspired by [21], we alternatively train on the VoxCeleb2 dataset (2025 h of speech) [34].

For DIHARD experiments, we use the *chunking* initialization introduced in Section V.

C. Evaluation Metric

Diarization Error Rate (DER) as defined by NIST [35] is used to evaluate the systems. As is the standard practice (for both datasets), we use the oracle speech activity labels – we drop the silence parts from the signal – so that only speaker errors are accounted for in the DER, (there are no missed speech and false alarm speech errors).

To be able to compare results with other works presented for the CALLHOME dataset, we evaluate the systems applying the standard 250 ms forgiveness collar around speaker change points and we do not evaluate any overlapped speech (forgiving evaluation).

For DIHARD, we evaluate the system with no collar and account for the overlap speech (strict evaluation), as it was done in the DIHARD challenge [20]. Given that our system does not model overlapped speech, for the DIHARD dataset, it is all accounted as missed speech error. For all the DERs reported for the DIHARD dataset, an absolute 8.18% and 9.52% of the error is due to the missed overlapped speech on the development and evaluation sets, respectively.

VIII. ANALYSIS AND OPTIMIZATION OF THE DIARIZATION ALGORITHM PARAMETERS

A. Interaction of the Parameters

As pointed out in Section IV, there is a set of parameters which can control the performance of our diarization algorithm. More specifically, these parameters define the configuration and topology of our Bayesian HMM and control the VB inference in the model.

This section provides the reader with an analysis on the effect of these parameters, their interaction and the sensitivity of the diarization performance to the parameters.

We carry out this analysis on the CALLHOME dataset that provides a more straightforward interpretation as compared to the DIHARD dataset. A similar analysis on DIHARD would be more intricate as audios come from different sources, where the optimal setting of the parameters may be different for each such source.

Figs. 3 and 4, show contours of the DER as a function of the parameters P_{loop} vs $downsamplingFactor$, and *acoustic scaling factor* F_A vs *speaker regularization coefficient* F_B, respectively.[5] While the other parameters stay fixed at their default values $P_{\text{loop}} = 0.9$, $downsamplingFactor = 25$, $F_A = 0.2$ and $F_B = 1$. As can be seen in the figures, these default values are close to the optimal setting of the parameters (for the CALLHOME dataset). For all the plots, the VB inference was

[5]Figures showing the DER as a function of all possible pairs of parameters are available in [Online]. Available: http://www.fit.vutbr.cz/~mireia/DER_plots.pdf

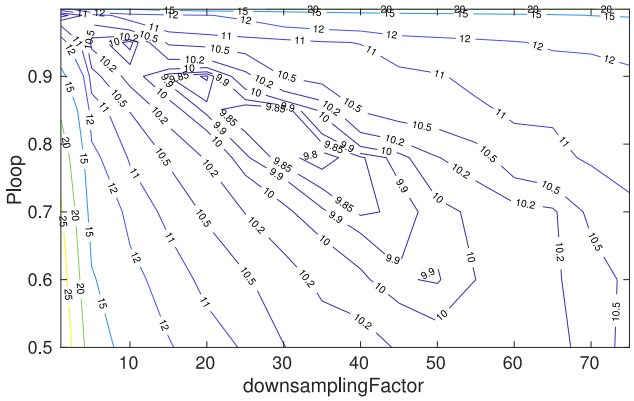

Fig. 3. P_{loop} versus $downsamplingFactor$.

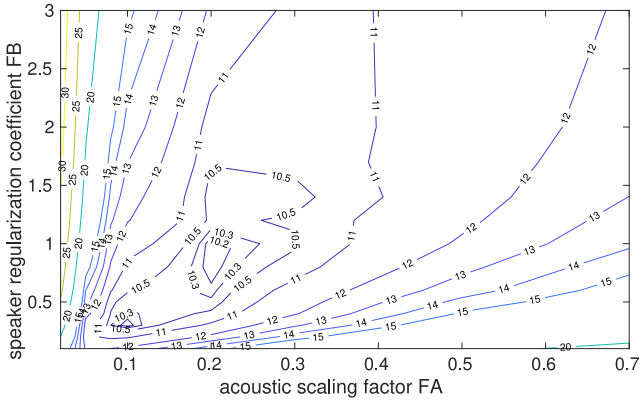

Fig. 4. F_B versus F_A.

initialized with an i-vector PLDA AHC system and no speaker merging was applied.

In the figures, linear scales are used for all the parameters. Although it may seem that a different scale may be more appropriate for some of them (e.g. logit scale for P_{loop}), we decided to keep the linear ones as we believe that they provide a better insight into the non-linear sensitivity of the parameters.

As can be seen in Fig. 3, the P_{loop} and the *downsamplingFactor* highly depend on each other. This is to be expected as both parameters participate in modeling the speaker turn duration (higher value of any of these parameters imposes longer speaker turns). In Fig. 4, we see that F_A and F_B are also quite interdependent, which is related to their interaction in the update formula (16). The remaining interactions of the parameters do not seem to be very significant when making only small changes around the optimal values.

In Section III-A the parameters F_A and F_B were introduced as factors scaling the first two terms in the modified ELBO (11). Therefore, their theoretically correct value leading to standard Variational Bayes inference is $F_A = F_B = 1$. Nevertheless, we see that the optimal value for the F_A is 0.2, which is necessary to compensate for the wrong assumption of statistical independence between observations as was explained in Section IV. On the other hand, in this CALLHOME based analysis, the optimal value of the F_B parameter is the theoretically correct $F_B = 1$. However, as we will see in Section VIII-C, other values may be better when training and evaluating on mismatched data.

TABLE II
RESULTS ON CALLHOME DATASET WITH AND WITHOUT SPEAKER MERGING CONSIDERING DIFFERENT INITIALIZATION METHODS

Initialization	No merge	Merge
i-vector PLDA AHC	9.89	9.82
rand x1	12.83	11.39
rand x5	8.89	8.66
chunking	13.42	9.67

B. Initializations and Speaker Model Merging

Table II compares the different initialization methods introduced in Section V and the speaker merging from Section VI. All results are again presented on the CALLHOME dataset and for the default parameters: $P_{\text{loop}} = 0.9$, $downsamplingFactor = 25$, $F_A = 0.2$ and $F_B = 1$.

The first column in Table II shows the results for different initializations (without speaker merging). We can see that both, the random initialization and the newly introduced *chunking*, achieve similar performance. The system initialized with the i-vector PLDA AHC achieves a significantly better result (9.89 %DER). Still, running the simple random initialization 5 times and selecting the result with the best ELBO provides the best performance (8.89 %DER). The second column of Table II shows the results for the same initializations but this time with speaker merging applied. The speaker merging approach provides a consistent gain in all cases. This improvement is more pronounced for the *chunking* initialization where the VB inference starts with very high number of speaker models. In this case, it is more likely to end up in a local optimum where speech of one speaker is attributed to multiple speakers (states) in the model and the speaker merging is designed to compensate for this problem. Results of the systems initialized with the i-vector PLDA AHC and *chunking* are now comparable after the merging of speakers (9.82 vs 9.67 %DER). Still, the best performance is obtained with the random x5 initialization achieving 8.66 %DER with the speaker merging. To our knowledge this is the best result published on CALLHOME with a system trained on publicly available data.

C. Effect of the Parameters for a Source-Diverse Dataset

The CALLHOME dataset used for the previous experiments contains quite clean speech, the number of speakers is the same in most of the recordings (2–3, see Section VII-A), and the system is trained with in-domain data. We therefore extend the analysis to the DIHARD dataset, to see the effect of the diarization algorithm parameters under more challenging conditions.

The DIHARD dataset contains audios coming from a variety of domains (see Table I), which differ in several aspects: they are recorded in noisy environments (e.g. VAST, CIR), are studio recordings (e.g. DCIEM, YP), contain child speech (e.g. ADOS, SEEDLINGS), etc. Also, the average number of speakers varies per domain, ranging from a single speaker (LIBRIVOX) to an average of 7 speakers (SCOTUS). DIHARD poses a nice scenario for testing the effects of the parameters.

In the following analysis, we will make use of the *chunking* initialization. On the CALLHOME dataset, we have shown

TABLE III
RESULTS ON DIHARD DATASET WITH PARAMETERS GLOBALLY TUNED ON THE DEV SET

Training set		No merge	Merge
DIHARD	**Dev**	17.32	15.98
	Eval	31.52	30.02
VoxCeleb	**Dev**	23.85	23.46
	Eval	29.76	28.85

that better performance can be attained with other initialization methods. However, the purpose of this analysis is not to achieve the best number on the DIHARD dataset; rather, we would like to provide a good insight into the behavior of the algorithm and interpretation of its parameters. The use of a more elaborate initialization can influence the behavior of the algorithm [21], and we aim to show the capabilities of the VB inference on its own. Also, unlike the external or random initializations, the *chunking* makes the results reproducible for anyone downloading our software [25].

Table III shows the optimal results that can be obtained on the DIHARD dataset after tuning all the parameters to their optimal values globally for all conditions. We show the results for systems trained on either DIHARD development set or VoxCeleb2 data (grey color indicates cheating results, where the system is trained and tested on the same data). We can see that training on VoxCeleb2 (2025 h) provides better results on the evaluation set, although the system trained on DIHARD data only (31 h for UBM, 14 h for matrix $\mathbf{V}$) is still competitive. As in previous experiments, the speaker merging provides consistent improvement in all cases. The optimal settings are different depending on the training set: $P_{\text{loop}} = 0.85$ and $F_A = 0.2$ for DIHARD training set, $P_{\text{loop}} = 0.6$ and $F_A = 0.1$ for VoxCeleb2 training set. The other two parameters are the default values: $downsamplingFactor = 25$, $F_B = 1$ in both cases.

Table IV shows the results obtained with the system trained on DIHARD data, when the parameters are optimized either globally or individually for each of the DIHARD domains.[6] Speaker merging is applied in all cases. The cheating results on the development set are again in grey. Of course, a practical system using per-domain optimized parameters would require source identification, as performed in [36]. The results are significantly improved with such per-domain optimization: DER goes from 30.02% to 27.10% on the eval set. The improvement is more significant for some of the domains e.g. DER in DCIEM evaluation set improves from 14.96% to 7.1%. Overall, domains with similar characteristics have the same optimal parameters, but the optimal parameters used per-domain can differ significantly: optimal P_{loop} values range from 0.3 (YP) to 0.85 (VAST). $F_B = 2$ is now the optimal value for SEEDLINGS. Note that LIBRIVOX can be *excluded* from the diarization task as an optimal setting will always result in finding a single speaker per file.

We also analyze the number of speakers found per recording. Fig. 5 shows histograms of the estimated number of speakers

[6]For CIR, which is only present in the evaluation set, we used the same parameter configuration as for RT04S, as we considered it the closest domain in the development set.

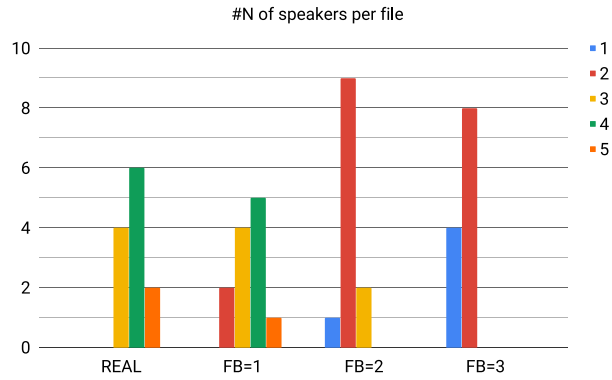

Fig. 5. Histogram of the real number of speakers per recording on the YP subset, and the histograms for the number of speakers found with different F_B values.

for the YP subset, when using different values of the speaker regularization coefficient F_B. It illustrates how higher values of F_B result in converging to fewer speakers per recording. Note that, besides seeking for the best DER (which is not always attained with the correct number of speakers), this parameter can be useful if, for any reason, we would like the system to over-cluster or under-cluster.

To complement the analysis of systems trained on in-domain DIHARD development set, Table V shows results for systems trained on the out-of-domain VoxCeleb2 dataset. Again, compared to using global parameter setting, the per domain parameter optimization carried out using the DIHARD development set provides an improvement on the DIHARD evaluation set (the overall DER improves from 28.85% to 27.37%). Looking at the optimal per-domain parameters, we again obtain a wide range of values for P_{loop} (from 0.5 to 0.9). On the other hand, the acoustic scaling $F_A = 0.1$ was consistently the best setting, which is lower than a typical value observed for in-domain training. Note that a smaller acoustic scaling F_A makes the model assume that the same amount of observations provide the inference with less evidence, which seems appropriate in the case of the mismatched model trained on the out-of-domain VoxCeleb2 data. A stronger regularization of speaker models seems to be also important in the case of out-of-domain training as the coefficient F_B larger than one was found optimal for more than half of the domains, with highest values $F_B = 3$ for SCOTUS and SEEDLINGS domains.

Even though the DIHARD dataset with its diversity of domains presents a good scenario for testing the utility of the diarization algorithm parameters, the dataset is quite small when considered the amounts of data per domain (up to only 12 files). Therefore, we reported only general trends observed in the experiments with the per-domain optimization. In order to make a more detailed and robust per-domain parameter analysis, we need to wait for a larger dataset collection.

IX. DEPENDENCY ON THE AMOUNT OF TRAINING DATA

In this section, we analyze the amount of data needed to train reliable models for diarization. The UBM and the $\mathbf{V}$ matrix used in the model have to be pre-trained. In the previous section,

TABLE IV
RESULTS WHEN OPTIMIZING PARAMETERS ON THE DEVELOPMENT SET FOR THE SYSTEM TRAINED ON DIHARD, USING *chunking* INITIALIZATION

	Optimization	ALL	ADOS	DCIEM	LIBRIV.	SCOTUS	SEEDLINGS	YP	VAST	RT04S	SLX	
Dev	Global	15.98	10.45	8.91	7.76	3.13	23.87	6.14	32.89	27.35	14.73	
	Per domain	13.87	7.77	5.87	0	3.04	22.21	2.57	30.69	27.35	14.85	
		ALL	ADOS	DCIEM	LIBRIV.	SCOTUS	SEEDLINGS	YP	VAST	ROAR	MIXER6	CIR
Eval	Global	30.02	22.27	14.96	9.00	16.67	48.19	13.38	42.43	39.26	11.06	61.33
	per domain	27.10	20.78	7.1	0	17.17	44.95	10.25	35.58	38.18	11.13	60.95

TABLE V
RESULTS WHEN OPTIMIZING PARAMETERS ON THE DEVELOPMENT SET FOR THE SYSTEM TRAINED ON VOXCELEB2, USING *chunking* INITIALIZATION

	Optimization	ALL	ADOS	DCIEM	LIBRIV.	SCOTUS	SEEDLINGS	YP	VAST	RT04S	SLX	
Dev	Global	23.46	25.94	7.23	10.08	11.15	46.50	8.37	31.12	38.33	27.54	
	per domain	20.32	25.94	6.95	0	8.08	39.98	4.11	30.82	37.44	21.76	
		ALL	ADOS	DCIEM	LIBRIV.	SCOTUS	SEEDLINGS	YP	VAST	ROAR	MIXER6	CIR
Eval	Global	28.85	27.18	8.15	4.39	15.04	55.01	7.62	37.15	40.06	9.12	62.91
	Per domain	27.37	27.18	8.75	0	14.69	47.90	6.00	34.55	41.71	7.86	62.24

we have shown results for systems trained on either VoxCeleb2 (2025 h of speech) or the DIHARD sets (31 h for UBM and 14 h for i-vector extractor training). In both cases, the systems perform similarly (see Tables IV and V). In this section, we analyze the performance of the system depending on the amount of training data available. For that purpose, we trained a system on the full VoxCeleb2 dataset (∼2000 h of speech), and systems on randomly selected subsets of 200 h, 20 h and 2 hours of speech.

Table VI shows that the system does not require large amounts of training data to perform competitively. The systems trained on ∼2000 h and 200 h perform similarly. In fact, 200 h provides better results than the full set, but we attribute the difference to noise in the results. The results for the systems trained on 20 h and 2 h show increasing degradation, but the systems still perform *reasonably* considering the little amount of data used for training. This insight is specially helpful for scenarios in which a diarization system would have to be trained with little (in-domain) data. Unlike actual state-of-the-art diarization systems based on x-vector extraction methods, which require loads of data to train competitive NNs, our model remains reliable when trained on only small amount of data.

Note also that, not only the parameters of the diarization algorithms from Section IV, but also the setting used for the UBM and the i-vector extractor (for the $\mathbf{V}$ matrix training) are the same for all the results presented in this section. The systems trained on smaller amounts of data could further benefit from readjusting these settings (e.g. smaller UBM).

TABLE VI
DER OF SYSTEMS USING TRAINING SETS OF DIFFERENT SIZES FOR THE UBM AND TMATRIX, FOR THE DIHARD DEV AND EVAL DATASETS

Training set	All VoxCeleb2	200h	20h	2h
Dev	23.46	23.27	25.01	27.81
Eval	28.85	28.61	31.38	32.51

X. COMPLEXITY OF THE ALGORITHM AND ITS EFFICIENT IMPLEMENTATION

This section comments on the time and memory complexity of the complete diarization algorithm. Suggestions for its efficient implementation are also provided, which should facilitate understanding of our publicly available python code [25]. To gain some intuition about the actual speed of the algorithm, we report time spent on the individual steps of the algorithm as measured for our (reasonably optimized) python implementation on *single core* of Intel Xeon E5-2680 v4 running at 2.40 GHz. For this purpose, the algorithm is run using our DIHARD setup (see Section VII-B), where the parameters mainly affecting the speed and memory requirements are set as follows: number of Gaussian components in GMM-UBM $C = 1024$, rank of the eigenvoice matrix $R = 400$, feature dimensionality $D = 40$, $downsamplingFactor = 25$.

For a time efficient implementation of our diarization algorithm, it is necessary to pre-calculate and store the matrices $\mathbf{V}_c^T \boldsymbol{\Sigma}_c^{\mathrm{ubm}^{-1}} \mathbf{V}_c$ from (14). It requires storing CR^2 numbers, which usually dominates the memory requirements of the algorithm.

For each input recording, the UBM-GMM is first evaluated and used to collect the sufficient statistics ζ_{tc} and $\boldsymbol{\rho}_t$ using (12) and (13). This step needs to be performed only once for each recording. It takes about 20 ms per 1 s of input speech to compete this step with our implementation and setup (i.e. about 50 times faster than real-time). We use a sparse matrix to store only the non-negligible responsibilities ζ_{tc} (about 1% of the values), which not only leads to memory savings, but also greatly speeds up the collection of the first order statistics $\boldsymbol{\rho}_t$ and calculation of other quantities that depend on ζ_{tc}. The first order statistics $\boldsymbol{\rho}_t$ are stored for all frames as $R \times T$ matrix, where T is number of speech frames. This matrix, which is repeatedly used in the VB iterations, might dominate the memory requirements for long conversations (i.e. hours of speech).

The time complexity of the following iterative VB inference depends linearly on the number of the iterations used and the number of speakers S considered in the model. On the other hand, the algorithm can be speed up $downsamplingFactor$ times using this parameter. With our tipicall setting of $downsamplingFactor = 25$, it takes about 1.5 ms per 1 speaker and 1 s of input speech to complete one VB iteration. The inference usually converges in less than 10 iterations. Therefore, running the diarization for 10 iterations with

model (for example randomly initialized) for 10 speakers takes $20 + 10 \times 10 \times 1.5 = 170$ ms, which corresponds to about 6 times faster than realtime on single CPU core.

The VB updates (specifically equations (18) and (23)) involve the second order statistics $\mathbf{\Phi}_t$. Using (14) to explicitly evaluate and store the statistics for every frame would be very time and memory consuming. Fortunately, this is not necessary. We can directly substitute (14) into (18) and rearrange the terms to obtain the speaker model precision as

$$\mathbf{L}_s = \mathbf{I} + \frac{F_A}{F_B} \sum_c \mathbf{V}_c^T \mathbf{\Sigma}_c^{\mathrm{ubm}^{-1}} \mathbf{V}_c \sum_t \gamma_{ts} \zeta_{tc}, \tag{26}$$

which can be very efficiently evaluated using the pre-calcualted quatities $\mathbf{V}_c^T \mathbf{\Sigma}_c^{\mathrm{ubm}^{-1}} \mathbf{V}_c$.

In (23), the second order statistics $\mathbf{\Phi}_t$ appear term $\mathrm{tr}(\mathbf{\Phi}_t \mathbf{S}_s)$, where we have defined $\mathbf{S}_s = \mathbf{L}_s^{-1} + \boldsymbol{\alpha}_s \boldsymbol{\alpha}_s^T$. This term can be efficiently evaluated as

$$\begin{aligned} \mathrm{tr}\left(\mathbf{\Phi}_t \mathbf{S_s}\right) &= \sum_c \zeta_{tc} \mathrm{tr}\left(\mathbf{V}_c^T \mathbf{\Sigma}_c^{\mathrm{ubm}^{-1}} \mathbf{V}_c \mathbf{S_s}\right) \\ &= \sum_c \zeta_{tc} \mathrm{vec}\left(\mathbf{V}_c^T \mathbf{\Sigma}_c^{\mathrm{ubm}^{-1}}\right)^T \mathrm{vec}\left(\mathbf{S_s} \mathbf{V}_c^T\right), \end{aligned} \tag{27}$$

where operator $\mathrm{vec}(\cdot)$ corresponds to vectorization of a matrix. In fact, evaluation of this term is the most time consuming operation in the VB iterations (about 90% of time is spent on performing the calculations from the second row of (27)).

XI. Conclusion

In this paper, we have presented a complete analysis of the Variational Bayes speaker diarization system with HMMs. The derivation of the formulas for the inference in the model were provided, and a detailed study was made to understand how to optimize the parameters that controls the diarization algorithm.

The use of the newly introduced *speaker regularization coefficient* has proven useful to control the algorithm for converging to different number of speakers per utterance. An effective naive initialization method has also been used, which makes the diarization algorithm competitive without the need of an external diarization method (nor extra training data), at the expense of making the diarization process slower. The speaker merging strategy was introduced as an effective way of avoiding local optima although, once again, it slows down the algorithm considerably.

Future work will explore ways of tuning the optimal parameter settings automatically based on test data. Also, we will seek for ways of implementing the speaker model merging in a more efficient way.

Appendix A
Evaluating the Likelihood for the Speaker Models

Given a Gaussian Mixture Model

$$p(\mathbf{x}) = \sum_c p(c) p(\mathbf{x}|c) = \sum_c w_c \, \mathcal{N}(\mathbf{x}_c | \boldsymbol{\mu}_c, \mathbf{\Sigma}_c) \tag{28}$$

and an arbitrary categorical distribution over its components $q(c)$, we can write

$$\begin{aligned} \ln p(\mathbf{x}) &= \sum_c q(c) \ln \left(\frac{p(\mathbf{x}|c) p(c) q(c)}{p(c|\mathbf{x}) q(c)} \right) \\ &= \sum_c q(c) \ln p\left(\mathbf{x}|c\right) - D_{KL}\left(q(c) \| p(c)\right) \\ &\quad + D_{KL}\left(q(c) \| p(c|\mathbf{x})\right), \end{aligned} \tag{29}$$

where $D_{KL}(q\|p)$ denotes Kullback-Leibler (KL) divergence between distributions q and p. Now, in (30) (at the bottom of this page), we apply the same manipulation to express the speaker-specific log likelihoods $\ln p(\mathbf{x}|\mathbf{y}_s)$ while setting $q(c) := p_{\mathrm{ubm}}(c|\mathbf{x}) = \zeta_c$. The last term in the first row of (30) is $-D_{KL}(q(c)\|p_{\mathrm{ubm}}(c))$, which corresponds to the first KL divergence term from (29). Note, however, that the second KL divergence term is missing in (30) for the following reason: To simplify the inference in our model, we assume that the responsibilities of speaker-specific Gaussian components for generating frame $\mathbf{x}$ are $p(c|\mathbf{x}, \mathbf{y}_s) = p_{\mathrm{ubm}}(c|\mathbf{x})$. In other words, we assume that the (probabilistic) alignment of frames to Gaussian components can be calculated using the UBM instead of the speaker-specific models. This is a reasonable assumption as we have the correspondence between the Gaussian components of the UBM and the speaker-specific GMMs. Note that the

$$\begin{aligned} \ln p(\mathbf{x}_t|\mathbf{y}_s) &= \ln \left(\sum_c w_c^{\mathrm{ubm}} \mathcal{N}\left(\mathbf{x}_t | \mathbf{m}_c^{\mathrm{ubm}} + \mathbf{V}_c \mathbf{y}_s, \mathbf{\Sigma}_c^{\mathrm{ubm}}\right) \right) = \sum_c \boldsymbol{\zeta}_{tc} \ln \mathcal{N}\left(\mathbf{x}_t | \mathbf{m}_c^{\mathrm{ubm}} + \mathbf{V}_c \mathbf{y}_s, \mathbf{\Sigma}_c^{\mathrm{ubm}}\right) + \sum_c \zeta_{tc} \ln \frac{w_c^{\mathrm{umb}}}{\zeta_{tc}} \\ &= \underbrace{-\frac{D}{2} \ln 2\pi - \sum_s \frac{\boldsymbol{\zeta}_{tc}}{2} \ln |\mathbf{\Sigma}_c^{\mathrm{ubm}}| - \frac{1}{2} \sum_s \boldsymbol{\zeta}_{tc} \left(\mathbf{x}_t - \mathbf{m}_c^{\mathrm{ubm}}\right)^T \mathbf{\Sigma}_c^{\mathrm{ubm}^{-1}} \left(\mathbf{x}_t - \mathbf{m}_c^{\mathrm{ubm}}\right) + \sum_c \zeta_{tc} \ln \frac{w_c^{\mathrm{umb}}}{\zeta_{tc}}}_{G(\mathbf{x}_t)} \\ &\quad + \mathbf{y}_s^T \underbrace{\sum_c \boldsymbol{\zeta}_{tc} \mathbf{V}_c^T \mathbf{\Sigma}_c^{\mathrm{ubm}^{-1}} \left(\mathbf{x}_t - \mathbf{m}_c^{\mathrm{ubm}}\right)}_{\boldsymbol{\rho}_t} - \frac{1}{2} \mathrm{tr}\Bigg(\mathbf{y}_s \mathbf{y}_s^T \underbrace{\sum_c \boldsymbol{\zeta}_{tc} \mathbf{V}_c^T \mathbf{\Sigma}_c^{\mathrm{ubm}^{-1}} \mathbf{V}_c}_{\mathbf{\Phi}_t} \Bigg) \\ &= G(\mathbf{x}_t) + \mathbf{y}_s^T \boldsymbol{\rho}_t - \frac{1}{2} \mathrm{tr}\left(\mathbf{y}_s \mathbf{y}_s^T \mathbf{\Phi}_t\right) \end{aligned} \tag{30}$$

same assumption is also made by similar models for speaker recognition (e.g. i-vectors extraction [1] or JFA [2]). Under such assumption, the missing term is the KL divergence between the same distributions $q(c) = p_{\text{ubm}}(c|\mathbf{x})$ and therefore it evaluates to zero. In practice, the distributions $p(c|\mathbf{x}, \mathbf{y}_s)$ and $p_{\text{ubm}}(c|\mathbf{x})$ will be similar but not exactly the same. Therefore, strictly speaking, (30) is only an approximation (lower bound) to the true log likelihood $\ln p(\mathbf{x}|\mathbf{y}_s)$.

Alternatively, we could further alleviate this approximation by extending our Bayesian model with additional latent variables c_{st} defining speaker-specific assignment of frames to Gaussian components. However, this would make the inference in the model much more computationally expensive. The statistics (12)–(14) would become speaker-dependent and would have to be extracted for each speaker and each frame using speaker-specific responsibilities $q(c_{st})$.

Appendix B
Derivation of the Update Formulas

Using (16) and (30), we can derive expected likelihood of an observation given a speaker, which will be useful in the following derivations:

$$\begin{aligned} &E_{q(\mathbf{Y})}\left[F_A \ln p(\mathbf{x}_t|s)\right] \\ &\quad = E_{q(\mathbf{y}_s)}\left[F_A \ln p(\mathbf{x}_t|s)\right] \\ &\quad = F_A\left[G(\mathbf{x}_t) + \boldsymbol{\alpha}_s^T\boldsymbol{\rho}_t - \frac{1}{2}\text{tr}\left(\left[\mathbf{L}_s^{-1} + \boldsymbol{\alpha}_s\boldsymbol{\alpha}_s^T\right]\boldsymbol{\Phi}_t\right)\right] \\ &\quad = \ln \overline{p}(\mathbf{x}_t|s) \end{aligned} \tag{31}$$

In order to derive the update formulas for the approximate posterior distributions, we maximize the modified ELBO (11) with respect to the distributions $q(\mathbf{Y})$ or $q(\mathbf{Z})$. This problem can be seen as a constrained optimization, where $q(\mathbf{Y})$ and $q(\mathbf{Z})$ are constrained to be valid probability density functions. To maximize the modified ELBO w.r.t $q(\mathbf{Y})$ (given fixed $q(\mathbf{Z})$), we construct the corresponding Lagrangian and set its functional derivative w.r.t $q(\mathbf{Y})$ equal to zero:

$$\frac{\partial}{\partial q(\mathbf{Y})}\left[\hat{\mathcal{L}}\left(q\left(\mathbf{Y},\mathbf{Z}\right)\right) + \lambda\left(\int q\left(\mathbf{Y}\right)d\mathbf{Y} - 1\right)\right] = 0 \tag{32}$$

Substituting (11) into (32), applying the functional derivative and solving for $q(\mathbf{Y})$ gives

$$\ln q(\mathbf{Y}) = \frac{F_A}{F_B}E_{q(\mathbf{Z})}\left[\ln p(\mathbf{X}|\mathbf{Y},\mathbf{Z})\right] + \ln p(\mathbf{Y}) + const. \tag{33}$$

where $const.$ is the constant term independent of $\mathbf{Y}$. Expanding (33) using the corresponding terms from (8), evaluating the expectation $E_{q(\mathbf{Z})}[.]$ w.r.t. current $q(\mathbf{Z})$ and using an additional manipulation and simplifying results in (induced) factorization

$$\ln q(\mathbf{Y}) = \sum_s \ln q(\mathbf{y}_s), \tag{34}$$

where the speaker-specific approximate log posteriors maximizing the modified ELBO are

$$\begin{aligned} \ln q(\mathbf{y}_s) &= \frac{F_A}{F_B}E_{q(\mathbf{Z})}\left[\sum_t \ln p(\mathbf{x}_t|z_t)\right] + \ln p(\mathbf{y}_s) + const. \\ &= \frac{F_A}{F_B}\sum_t \gamma_{ts}\ln p(\mathbf{x}_t|\mathbf{y}_s) + \ln p(\mathbf{y}_s) + const. \\ &= \frac{F_A}{F_B}\mathbf{y}_s^T\sum_t \gamma_{ts}\boldsymbol{\rho}_t \\ &\quad - \frac{1}{2}\text{tr}\left(\mathbf{y}_s\mathbf{y}_s^T\left[\frac{F_A}{F_B}\sum_t \gamma_{ts}\boldsymbol{\Phi}_t + \mathbf{I}\right]\right) + const., \end{aligned} \tag{35}$$

where the responsibilities $\gamma_{ts} = q(z_t = s)$ are the approximate marginal posteriors of the latent variables z_t derived from the current distribution $q(\mathbf{Z})$ (see below). Since $\ln q(\mathbf{y}_s)$ is a quadratic function (the log of a Gaussian distribution), completing the squares gives the final update formulas (16) to (18).

To maximize the modified ELBO w.r.t $q(\mathbf{Z})$ (given fixed $q(\mathbf{Y})$), we solve an equation similar to (32), where symbols $\mathbf{Y}$ and $\mathbf{Z}$ are exchanged. This time, solving for $q(\mathbf{Z})$ leads to

$$\begin{aligned} \ln q(\mathbf{Z}) &= F_A E_{q(\mathbf{Y})}\left[\ln p(\mathbf{X}|\mathbf{Y},\mathbf{Z})\right] + \ln p(\mathbf{Z}) + const. \\ &= F_A E_{q(\mathbf{Y})}\left[\sum_t \ln p(\mathbf{x}_t|z_t)\right] + \ln p(\mathbf{Z}) + const. \\ &= \sum_t \ln \overline{p}(\mathbf{x}_t|z_t) + \ln p(\mathbf{Z}) + const., \end{aligned} \tag{36}$$

where (31) was used to obtain the last line. Notice that the last line of equation (36) has exactly the same form as the posterior probability of the latent sequence $p(\mathbf{Z}|\mathbf{X})$ for the standard (non Bayesian) HMM except that the standard emission probabilities $p(\mathbf{x}_t|z_t)$ are replaced by $\overline{p}(\mathbf{x}_t|z_t)$. Therefore, to evaluate the marginals $\gamma_{ts} = q(z_t = s)$, we can use the same forward-backward algorithm as used in the standard HMM training for evaluating the responsibilities $p(z_t = s|\mathbf{X})$, using the quantities $\overline{p}(\mathbf{x}_t|z_t)$ instead of $p(\mathbf{x}_t|z_t)$. Equations (19) to (23) correspond to such modified forward-backward algorithm.

Appendix C
Derivation of the Lower Bound

The first term of the modified ELBO (11) can be evaluated (using (31)) as

$$\begin{aligned} &F_A E_{q(\mathbf{Y}),q(\mathbf{Z})}\left[\ln p(\mathbf{X}|\mathbf{Y},\mathbf{Z})\right] \\ &\quad = F_A E_{q(\mathbf{Y}),q(\mathbf{Z})}\left[\sum_t \ln\ p(\mathbf{x}_t|z_t)\right] \\ &\quad = E_{q(\mathbf{Z})}\left[\sum_t \ln \overline{p}(\mathbf{x}_t|z_t)\right] = \sum_t\sum_s \gamma_{ts}\ln\overline{p}(\mathbf{x}_t|s) \end{aligned} \tag{37}$$

Using the factorization (34), the second term of the ELBO (11) (excluding the scalar F_B) can be evaluated as

$$E_{q(\mathbf{Y})}\left[\ln\frac{p(\mathbf{Y})}{q(\mathbf{Y})}\right] = -\sum_s D_{KL}(q(\mathbf{y}_s)\|p(\mathbf{y}_s)) = \sum_s \frac{1}{2}\left(R + \ln|\mathbf{L}_\mathbf{s}^{-\mathbf{1}}| - tr(\mathbf{L}_s^{-1}) - \boldsymbol{\alpha}_s^T\boldsymbol{\alpha}_s\right), \tag{38}$$

which is negative sum of well-known KL divergences between pairs of Gaussian distributions. Finally, the third term in (11) is the negative KL divergence

$$E_{q(\mathbf{Z})}\left[\ln\frac{p(\mathbf{Z})}{q(\mathbf{Z})}\right] = \sum_{s=1}^{S}\gamma_{1s}\ln\frac{\pi_s}{\gamma_{1s}} + \sum_{t=2}^{T}\sum_{m=1}^{S}\sum_{n=1}^{S}\xi_{\mathrm{tmn}}\ln\frac{p(n|m)}{q(z_t=n|z_{t-1}=m)}, \tag{39}$$

where the approximate marginal probability of transitioning from state m to state n at time t

$$\xi_{\mathrm{tmn}} = q(z_{t-1}=m, z_t=n) = \frac{A(t-1,m)\bar{p}(\mathbf{x}_t|n)p(n|m)B(t,n)}{\bar{p}(\mathbf{X})} \tag{40}$$

can be estimated using the forward-backward algorithm (19) to (23) and where the approximate posterior of transitioning to state n at time t given previous state m

$$q(z_t=n|z_{t-1}=m) = \frac{\xi_{\mathrm{tmn}}}{\sum_s \xi_{\mathrm{tms}}}. \tag{41}$$

REFERENCES

[1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[2] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1435–1447, May 2007.

[3] P. Kenny, "Bayesian analysis of speaker diarization with eigenvoice priors," CRIM, Montreal, QC, Canada, Tech. Rep., 2008. [Online]. Available: https://www.crim.ca/perso/patrick.kenny/

[4] E. B. Fox, E. B. Sudderth, M. Jordan, and A. S. Willsky, "The sticky HDP-HMM: Bayesian nonparametric hidden Markov models with persistent states," MIT LIDS, Tech. Rep. P-2777, 2007.

[5] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 5, pp. 1557–1565, Sep. 2006.

[6] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 356–370, Feb. 2012.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York, NY, USA: Springer-Verlag, 2006.

[8] M. A. Siegler, U. Jain, B. Raj, and R. M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio," in *Proc. DARPA Speech Recognit. Workshop*, 1997, pp. 97–99.

[9] S. E. Johnson, "Who spoke when? - automatic segmentation and clustering for determining speaker turns," in *Proc. EUROSPEECH*, 1999, vol. 5, pp. 2211–2214.

[10] H. Jin, F. Kubala, and R. Schwartz, "Automatic speaker clustering," in *Proc. DARPA Speech Recognit. Workshop*, Feb. 1997, pp. 108–111.

[11] S. S. Chen and P. S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian information criterion," in *Proc. DARPA Broadcast News Transcript. Understand. Workshop*, 1998, pp. 127–132.

[12] J. Gauvain, L. Lamel, and G. Adda, "Partitioning and transcription of broadcast news data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1998, pp. 1335–1338.

[13] C. Barras, X. Zhu, S. Meignier, and J. Gauvain, "Improving speaker diarization," in *Proc. Fall Rich Transcript. Workshop*, 2004.

[14] F. Castaldo *et al.*, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2008, pp. 4133–4136.

[15] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA I-vector scoring and unsupervised calibration," in *Proc. IEEE Spoken Lang. Technol. Workshop*, Dec. 2014, pp. 413–417.

[16] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE 11th Int. Conf. Comput. Vision*, Oct. 2007, pp. 1–8.

[17] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. Odyssey Speaker Lang. Recognt. Workshop*, Brno, Czech Republic, Jun. 2010. [Online]. Available: https://www.crim.ca/perso/patrick.kenny/kenny_Odyssey2010.pdf

[18] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2016, pp. 165–170.

[19] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2017, pp. 4930–4934.

[20] N. Ryant *et al.*, "First DIHARD challenge evaluation plan," 2018. [Online]. Available: https://zenodo.org/record/1199638

[21] G. Sell *et al.*, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD Challenge," in *Proc. Interspeech*, 2018, pp. 2808–2812.

[22] M. Diez *et al.*, "BUT system for DIHARD speech diarization challenge 2018," in *Proc. Interspeech*, 2018, pp. 2798–2802.

[23] F. Valente, P. Motlicek, and D. Vijayasenan, "Variational Bayesian speaker diarization of meeting recordings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 4954–4957.

[24] Fabio Valente, "Variational Bayesian methods for audio indexing," Ph.D. dissertation, 2005.

[25] L. Burget, "VB diarization with eigenvoice and HMM priors," 2013. [Online]. Available: http://speech.fit.vutbr.cz/software/vb-diarization-eigenvoice-and-hmm-priors

[26] M. Diez, L. Burget, and P. Matějka, "Speaker diarization based on Bayesian HMM with eigenvoice priors," in *Proc. Odyssey Speaker Lang. Recognit. Workshop*, 2018, pp. 147–154.

[27] A. F. Martin and M. A. Przybocki, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *Proc. 7th Eur. Conf. Speech Commun. Technol.*, Sep. 2001, vol. 7, pp. 787–790.

[28] N. Ryant d *et al.*, "DIHARD corpus. Linguistic data consortium.," 2018. [Online]. Available: https://coml.lscp.ens.fr/dihard/2018/instructions.html

[29] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, , Gatsby Unit, University College London, London, U.K., 2003.

[30] G. Sell and D. Garcia-Romero, "Diarization resegmentation in the factor analysis subspace," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 4794–4798.

[31] "NIST SRE 2000 evaluation plan," [Online]. Available: https://www.nist.gov/sites/default/files/documents/2017/09/26/spk-2000-plan-v1.0.htm_.pdf

[32] P. Matějka *et al.*, "Full-covariance UBM and heavy-tailed PLDA in I-vector speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 4828–4831.

[33] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 7, pp. 1717–1731, Sep. 2010.

[34] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[35] "NIST rich transcription evaluations," [Online]. Available: https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation, version: md-eval-v22.pl

[36] Z. Zajíc, M. Kunešová, M. Hrúz, and J. Vaněk, "UWB-NTIS speaker diarization system for the DIHARD II 2019 challenge," in *Proc. Interspeech*, 2019, pp. 993–997.