OPTIMIZING BAYESIAN HMM BASED X-VECTOR CLUSTERING FOR THE SECOND DIHARD SPEECH DIARIZATION CHALLENGE

Mireia Diez[1,⋆], *Lukáš Burget*[1,⋆], *Federico Landini*[1], *Shuai Wang*[1,2], *Honza Černocký*[1]

[1]Brno University of Technology, Faculty of Information Technology, IT4I Centre of Excellence, Czechia
[2] Speechlab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
{mireia, burget, landini, cernocky}@fit.vutbr.cz; feixiang121976@sjtu.edu.cn

ABSTRACT

This paper presents an analysis of our diarization system winning the second DIHARD speech diarization challenge, track 1. This system is based on clustering x-vector speaker embeddings extracted every 0.25s from short segments of the input recording. In this paper, we focus on the two x-vector clustering methods employed, namely Agglomerative Hierarchical Clustering followed by a clustering based on Bayesian Hidden Markov Model (BHMM). Even though the system submitted to the challenge had further post-processing steps, we will show that using this BHMM solely is enough to achieve the best performance in the challenge. The analysis will show improvements achieved by optimizing individual processing steps, including a simple procedure to effectively perform "domain adaptation" by Probabilistic Linear Discriminant Analysis model interpolation. All experiments are performed in the DIHARD II evaluation framework.

Index Terms— Speaker Diarization, Variational Bayes, HMM, x-vector, DIHARD

1. INTRODUCTION

In our previous works, we introduced a Bayesian Hidden Markov Model (BHMM) with eigenvoice priors for diarization (often referred to as Variational Bayes (VB) diarization) [1, 2]. Recently, we introduced a simplified variant of the BHMM for a fast and robust clustering of x-vectors [3], which is in contrast to the original approach where the model was used to cluster frame-level Cepstral features. To robustly discriminate between speakers the BHMM uses Probabilistic Linear Discriminant Analysis (PLDA) pre-trained on x-vectors. The BHMM is used to model the sequence of neural network based speaker embeddings (x-vectors) that are extracted from consecutive short sub-segments of the input recording. The inference in this model gives us the diarization output, that is, it infers the number of speakers, the speaker models and the assignment of x-vectors to speakers.

Experiments performed on the DIHARD I dataset [4, 5] showed that this model outperforms the Agglomerative Hierarchical Clustering (AHC) commonly used to cluster x-vectors for diarization tasks [6, 1]. In this work, we further analyze the model and show that, using more appropriate configurations, the BHMM would achieve the best performance in the recent DIHARD II challenge (track 1) even without the need of re-segmentation or overlap speech post-processing steps that were present in our official winning submission to this challenge. We further show how to effectively adapt BHMM to the target domain by a simple interpolation of PLDA models.

This work is conducted on the DIHARD II dataset [7], designed for the second of a yearly evaluation series focusing on hard diarization conditions. The Second DIHARD Speech Diarization Challenge had four different tracks: tracks one and two focused on single-channel diarization following DIHARD I format; tracks three and four introduced multi-channel diarization tasks, using CHiME 5 data [8]. The present work focuses on the optimization of the BHMM system, which is the core of the system submitted to track 1. For more details on the whole system description for track 1 and description of the systems submitted by BUT team to the other tracks we refer the reader to [9].

2. BAYESIAN HMM FOR X-VECTOR CLUSTERING

The diarization model used for x-vector clustering is a Bayesian HMM, where the states correspond to speakers, the transition between states represent the speaker turns and the state distributions are derived from a PLDA model pre-trained on labeled x-vectors in order to facilitate discrimination between speaker voices.

The HMM has a one-to-one correspondence between the HMM states and speakers. The HMM model is ergodic (i.e. transitions between all speakers are possible), where the (initial) number of states is chosen to be at least the highest number of speakers we would expect to appear in a conversation.

The HMM topology and transition probabilities model the speaker turn durations (see Figure 1 for an example with 3 speaker states): P_{loop} is a tunable parameter, which corresponds to the probability of staying in the same state

⋆Equal contribution.

The work was supported by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

(speaker). For each input observation (x-vector), we leave the current state with probability $1-P_{\text{loop}}$ and we transition to new state of speaker s with probability π_s[1]. The probabilities π_s also control the selection of the initial HMM state. The probabilities π_s are inferred from the input conversation using the iterative VB updates. Thanks to the automatic relevance determination (ARD) [10, 2] principle, the values π_s for any redundant speakers tend to converge to zero. This allows to infer the number of speakers in the input conversation.

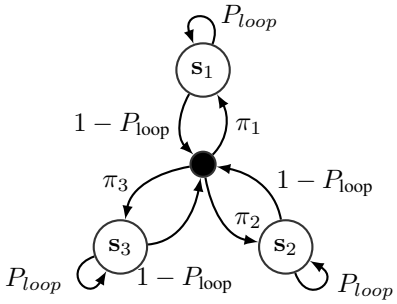

Fig. 1. HMM model for 3 speakers with 1 state per speaker, with a dummy non-emitting (initial) state.

Each speaker (or HMM state) specific distribution is a Gaussian modeled using PLDA like model, which is pre-trained on a large number of speaker labeled x-vectors. We consider the simplified PLDA model [11], which assumes that the distribution of speaker means

$$\mathbf{m}_s = \mathbf{m} + \mathbf{V}\mathbf{y}_s, \tag{1}$$

where $\mathbf{y}_s$ is a latent vector with standard normal prior and where the speaker specific distribution of x-vectors

$$p(\mathbf{x}_t|\mathbf{y}_s) = \mathcal{N}(\mathbf{x}_t; \mathbf{m}_s, \mathbf{\Sigma}_{wc}), \tag{2}$$

where $\mathbf{\Sigma}_{wc}$ is the within speaker covariance matrix shared by all speaker models. To estimate the speaker specific distributions, we only need to infer the $\mathbf{y}_s$ vectors.

To address the speaker diarization task for a given input observation (x-vector) sequence, the speaker distributions and the probabilistic alignment of observations to HMM states (speakers) are jointly estimated together with the π_s probabilities. We use Variational Bayes (VB) inference [10] to iteratively estimate the distributions using the update formulas that can be found in [3]. The most likely alignment of observations to HMM states (speaker) is the diarization solution. For a more detailed description of the model, update formulas, and definition of the configuration parameters, we kindly refer the reader to [1, 2, 3] or directly to our python implementation of this inference [12].

3. SYSTEM DESCRIPTION

The diarization system submitted to the track 1 of the second DIHARD speech diarization challenge consists of the following steps. In the training phase, an x-vector extractor and an i-vector extractor are trained. Also, two PLDA models are trained: one on x-vectors from the VoxCeleb dataset [13] and one on x-vectors from the DIHARD development set. The two models are interpolated by averaging the covariance matrices to obtain the final robust and domain adapted PLDA model, which is used for AHC and BHMM-based x-vector clustering. In the diarization phase, Weighted Prediction Error (WPE) [14] is used first to de-reverberate the speech signal. Then x-vectors are extracted from the input conversation using a 1.5s sliding window and a shift of 0.25s. The x-vectors are centered, whitened and length normalized (which is also done for the PLDA training data). Next the x-vectors are pre-clustered using AHC to obtain initial labels for the BHMM clustering. x-vectors are then clustered using the BHMM model. To refine the diarization output another BHMM model is used at frame-level (using MFCCs features) as re-segmentation step. This model uses the pre-trained i-vector extractor as part of the frame-level BHMM to model speaker distributions [2] (not to extract any i-vectors). Finally, the segments with overlapped speech of two-speakers are detected and post-processed to get two speaker labels assigned. This last overlapped speech detection step is not analyzed in this paper. Details on this step can be found in [9].

3.1. x-vector extraction

Our x-vector extractor is based on time-delay neural network (TDNN) similar to [15]. It is trained for speaker classification on VoxCeleb [13] training and VoxCeleb2 [16] development data with data augmentation, amounting to 6 million utterances from 7146 speakers. The utterances are further cut into 2s segments for the neural network training. 64-dimensional filter banks (Fbanks) are used as input features, using an energy-based voice activity detector (VAD) to remove silence. For test conversations, the 512 dimensional x-vectors are extracted from the penultimate layer every 0.25s from (up to) 1.5s segments. x-vectors are centered and whitened using statistics estimated from DIHARD development and evaluation data, and then length normalized.

3.2. PLDA models

The out-of-domain PLDA model is trained using the large VoxCeleb training set. The in-domain PLDA model is trained on the limited DIHARD dev set. Both models are estimated from centered, whitened and length-normalized x-vectors extracted from 3s segments. In our submission to DIHARD II challenge, we used the following domain adaptation strategy: The directions where the within- and across-class variability were higher in the in-domain PLDA model than in the out-of-domain PLDA were identified and the extra variability was added to the corresponding covariance matrices in the out-of-domain PLDA. Later, we found that the simple interpolation of the two PLDA models is sufficient and even slightly improves the results. Hence, experiments in this paper are performed with the latter simpler approach resulting in a slightly better performance than reported in the official leaderboard.

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

3.3. AHC

The AHC approach mainly follows [6] as implemented in [17] using Kaldi toolkit [18] with some optimizations. The full clustering process goes as follows: For each input conversation, x-vectors are extracted for 1.5s windows with 0.25s overlap (in contrast to the default 0.75s in [17]). A conversation specific Principal Component Analysis (PCA) is estimated on the x-vectors and used to project the x-vectors (and also PLDA model) to few dimensions so as to keep only a 30% of the total variability (instead of the Kaldi default 10%). See Section 5.1 for details on this new frame-rate and energy percentage settings. The projected x-vectors are once more length-normalized. PLDA speaker verification scores are calculated for pairs of x-vectors to construct the pairwise-similarity matrix that is used in the Unweighted Pair Group Method with Arithmetic Mean (UPGMA) variant of AHC. different-speaker hypothesis. The threshold used as stopping criteria for the AHC is fine-tuned on the development set. Before performing AHC, the scores in the similarity matrix are optionally calibrated using conversation specific unsupervised linear calibration as follows: a GMM with two univariate Gaussian components with shared variance is trained on all the scores from the similarity matrix. The two learned Gaussian components are assumed to model the distributions of same-speaker (the component with the higher mean) and different-speaker scores. The two Gaussian distributions and the corresponding learned priors (the component weights) are then used to map the scores in the similarity matrix to log odds ratios between the same- and different-speaker hypothesis.

3.4. BHMM clustering of x-vectors

The PLDA models used are the same as the ones trained for the AHC. The BHMM is initialized from the AHC diarization output. We tune the AHC to under-cluster so that the BHMM has more freedom to search for the optimal results and converge to the right number of speaker models[2]. The parameters of the model [3] are set as follows: Given that the input features are x-vectors extracted every 0.25s, the *downsamplingFactor* (commonly set to 25 for frame level features) and *min duration* are set to 1 and effectively not used. The *Speaker regularization coefficient* F_B was optimized on the development set and set to 11. Details on the *Acoustic scaling factor* F_A and P_{loop} can be found in the analysis in section 5.2.

3.5. Frame-level BHMM re-segmentation

As features, standard 19 MFCC plus Energy plus first order delta coefficients are used, extracted from 16kHz 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. The UBM-GMMs and the total variability matrix are trained using the VoxCeleb2 dataset, which amounts to 2025h of speech. A single iteration of this frame-level BHMM is applied [3]. The optimal values found for the parameters [2] are $F_A = 0.1$, $F_B = 1$, $P_{\text{loop}} = 0.95$, $min\ duration = 1$ and $downsamplingFactor = 5$.

[2]Note that the inference in BHMM cannot converge to higher number of speakers than what is suggested by the AHC-based initialization.

4. EVALUATION DATA AND METRIC

The experiments are evaluated on the DIHARD II dataset. This dataset was created for the second DIHARD challenge [7], the second of a yearly series of challenges designed to foster research on diarization in hard conditions. The dataset is an extension of the first DIHARD dataset [4] and includes utterances coming from several sources (YouTube, court rooms, meetings, etc.). The corpus consists of 192 development and 194 evaluation recordings, containing around 18h and 17h of speech, respectively.

The system is evaluated in terms of the Diarization Error Rate (DER) as defined by NIST [19]. We employ the format established for track 1 of the second DIHARD challenge: we use the oracle speech activity labels so that only speaker errors are accounted for in the DER, we evaluate the system with no collar and we evaluate the overlap speech regions.

5. SYSTEM ANALYSIS & RESULTS

5.1. AHC optimization

From previous works [3] and the Kaldi baseline [17], we inherited the practice of projecting x-vectors by means of PCA so as to keep only 10% of their variability (see section 3.3). Table 1 shows results of AHC clustering when the x-vectors are projected keeping 10%, 30% and 100% of variability and when the AHC is performed on non-calibrated or calibrated PLDA scores. As seen in the table, projecting the x-vectors to a smaller dimensionality is important to obtain better performance [20]. Still, for the DIHARD data, it is better to keep around 30% of the energy than only a 10%. Regarding calibration, it interestingly helps or harms depending on the percentage of energy kept. It improves performance when 10% energy is kept it hurts for the case of 100%. For the usually optimal PCA projection, keeping 30% of energy, calibration does not have a very significant effect, nevertheless, for further experiments, we decided to keep the 30% energy x-vectors and perform AHC on calibrated scores.

		% Energy kept		
Set	Calibration	10	30	100
Dev	No	21.33	**20.33**	23.44
	Yes	20.89	**20.46**	30.03
Eval	No	21.70	**21.19**	25.89
	Yes	**20.86**	21.12	30.98

Table 1. DER results attained with AHC using PLDA models trained on VoxCeleb with different energy levels kept for the x-vector projection and with/without score calibration

Set	VB reseg.	PLDA trained on VoxCeleb	DIHARD dev	Interp.
Dev	No	20.46	20.55	19.74
	Yes	19.84	20.20	**19.21**
Eval	No	21.12	22.29	**20.96**
	Yes	20.11	21.48	**19.97**

Table 2. DER results attained with AHC using PLDA models trained on VoxCeleb (out-of-domain), DIHARD dev (in-domain) and when interpolating them

In Table 2, we present results attained by the AHC when the PLDA is trained either on VoxCeleb data, on the DIHARD dev set or when both PLDA models are interpolated. In this table (and the following tables), numbers in gray denote cheating results, when the training data includes the test data. As it can be seen, even though by only a small margin, the interpolated model provides the best performance. Applying frame-level BHMM re-segmentation on top of the AHC output improves the results on eval set 1% absolute DER.

5.2. BHMM optimization

Table 3 shows the improvement that can be achieved by using x-vector level BHMM clustering. In order to obtain optimal results, the AHC used as initialization needs to be run so as to under-cluster the x-vectors (see section 3.4). Performing

Set	method	Threshold Optimal for AHC	Under-clustered
Dev	AHC	20.46	(33.55)
	BHMM	19.33	**18.34**
Eval	AHC	21.12	(33.31)
	BHMM	19.90	**19.14**

Table 3. DER for different clustering methods and thresholds

x-vector BHMM clustering provides a 2% absolute DER gain in both sets compared to the AHC (18.34% vs 20.46% in dev and 19.14% vs 21.12% on eval). Next, we analyze the effect

Set	F_A	P_{loop}	Frame rate 0.75s	0.25s
Dev	1.0	0.0	**19.55**	23.20
	0.4	0.8	20.13	**18.34**
Eval	1.0	0.0	**20.29**	22.89
	0.4	0.8	22.74	**19.14**

Table 4. DER for different x-vectors extracting frame rates

of using different frame rates for extracting the x-vectors. Table 4 compares results when extracting x-vectors every 0.75s (as done in [3]) or every 0.25s. We also illustrate which are the optimal settings for the F_A and P_{loop} parameters, as tunning them is crucial for obtaining best performance. Overall, increasing the frame rate for estimating x-vectors results in more than an 1% absolute DER gain in both dev and eval sets.

Set	VB reseg.	PLDA trained on Voxceleb	DIHARD dev	Interp.
Dev	No	18.34	17.87	17.90
	Yes	18.35	18.16	18.23
Eval	No	19.14	18.83	**18.39**
	Yes	18.95	18.80	**18.38**

Table 5. DER results attained with BHMM using PLDA models trained on VoxCeleb (out-of-domain), DIHARD dev (in-domain) and when interpolating them

In table 5, we analyze the effect of using the interpolated PLDA model for the BHMM x-vector clustering. The BHMM clustering benefits more than the AHC from the PLDA interpolation (see table 2), resulting in close to 0.7% absolute DER gain on the eval set.

Finally, table 5 presents also results after adding the BHMM re-segmentation step. Unfortunately, the frame-level BHMM re-segmentation provides only marginal gains after the x-vector level BHMM clustering. This is in contrast to the great gains seen in our previous participation [21] and when applied after AHC (see table 2). This is due to several factors: first of all, the x-vectors are newly extracted every 0.25s, which means 3 times better time resolution as compared to the typical 0.75s. Besides, the BHMM diarization output is better than that attained with the AHC, leaving less margin for improvements. Finally, the x-vector level BHMM clustering uses the interpolated PLDA model, adapted to the target domain, whereas the frame-level re-segmentation model is solely trained on (out of domain) VoxCeleb2 dataset (which also explains the small degradation obtained on the development set after the re-segmentation step). Our attempts to train the frame-level BHMM re-segmentation models on DIHARD dev data provided no significant gains on the evaluation so far. Still, this direction will be explored in the future.

6. CONCLUSIONS

In this work we have described the core of our winning system on track 1 of the second DIHARD speech diarization challenge, providing an analysis of the steps taken to optimize performance. We have shown that the improved x-vector extractor, increasing the frame-rate for x-vector extraction and using x-vector level BHMM diarization with PLDA model interpolation for "domain adaptation" significantly boosts performance.

Our last year's system [21] consisted of an x-vector extractor, AHC, frame-level BHMM diarization and overlapped speech detection modules, and (according to the leaderboard) attained 25.01% DER on the DIHARD II evaluation set. Compared to it, the actual described system improves performance on by close to an absolute 7% DER, attaining a (first position deserving) 18.39% DER. The code for this system is available in [22].

7. REFERENCES

[1] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on bayesian hmm with eigenvoice priors," in *Proceedings of Odyssey 2018, The speaker and Language Recognition Workshop*, 2018.

[2] M. Diez, L. Burget, F. Landini, and H. Černocký, "Analysis of speaker diarization based on bayesian hmm with eigenvoice priors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019. [Online]. Available: http://www.fit.vutbr.cz/~mireia/Diez19_Submitted_IEEE.pdf

[3] M. Diez, L. Burget, S. Wang, J. Rohdin, and H. Černocký, "Bayesian HMM based x-vector clustering for Speaker Diarization," in *Proc. Interspeech 2019*, 2019, pp. 346–350.

[4] N. Ryant and et al., "DIHARD Corpus. Linguistic Data Consortium." 2018.

[5] E. Bergelson, "Bergelson Seedlings HomeBank Corpus." 2016.

[6] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge," in *Interspeech*. ISCA, 2018, pp. 2808–2812.

[7] N. R. et. al., "The Second DIHARD Diarization Challenge: Dataset, task, and baselines." in *Proceedings of Interspeech 2016*, 2019.

[8] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'chime' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. of Interspeech 2020*, Hyderabad, India, Sep. 2018.

[9] F. Landini, S. Wang, M. Diez, L. Burget, P. Matějka, K. Žmolíková, L. Mošner, A. Silnova, O. Plchot, O. Novotný, H. Zeinali, and J. Rohdin, "BUT System for DIHARD Speech Diarization Challenge 2019," in *Submitted to Conference on Acoustics, Speech and Signal Processing (ICASSP) 2020*, 2020.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[11] N. Brummer, "EM for Simple PLDA," 2010, technical report.

[12] L. Burget, "VB Diarization with Eigenvoice and HMM Priors," http://speech.fit.vutbr.cz/software/vb-diarization-eigenvoice-and-hmm-priors, 2013, [Online; January-2017].

[13] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *INTERSPEECH*, 2017.

[14] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B. H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, Sept 2010.

[15] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *INTERSPEECH 2017*, August 2017, pp. 999–1003.

[16] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Proceedings of Interspeech 2018*, 2018, pp. 1086–1090. [Online]. Available: https://doi.org/10.21437/Interspeech.2018-1929

[17] Kaldi, "dihard_2018 v2," https://github.com/kaldi-asr/kaldi/tree/master/egs/dihard_2018/v2, [Downloaded: 2019-02].

[18] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[19] "NIST Rich Transcription Evaluations," https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation.

[20] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2015–2028, Oct 2013.

[21] M. Diez, F. Landini, L. Burget, J. Rohdin, A. Silnova, K. Žmolíková, O. Novotný, K. Veselý, O. Glembek, O. Plchot, L. Mošner, and P. Matějka, "But system for dihard speech diarization challenge 2018," in *Proc. Interspeech 2018*, 2018, pp. 2798–2802. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1749

[22] L. Burget, M. Diez, S. Wang, and F. Landini, "VBHMM x-vectors Diarization (aka VBx)," https://speech.fit.vutbr.cz/software/vbhmm-x-vectors-diarization.