INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Speaker Diarization using Leave-one-out Gaussian PLDA Clustering of DNN Embeddings

Alan McCree, Gregory Sell, Daniel Garcia-Romero

Human Language Technology Center of Excellence
Johns Hopkins University, Baltimore, MD, USA
alan.mccree@jhu.edu, gsell@jhu.edu, dgromero@jhu.edu

Abstract

Many modern systems for speaker diarization, such as the top-performing JHU system in the DIHARD 2018 challenge, rely on clustering of DNN speaker embeddings followed by HMM resegmentation. Two problems with this approach are that parameters need significant retuning for different applications, and that the DNN contributes only to the clustering task and not the resegmentation. This paper presents two contributions: an improved HMM segment assignment algorithm using leave-one-out Gaussian PLDA scoring, and an approach to training the DNN such that embeddings directly optimize performance of this scoring method with generatively updated PLDA parameters. Initial experiments with this new system are very promising, achieving state-of-the-art performance for two separate tasks (Callhome and DIHARD18) without any task-dependent parameter tuning.

Index Terms: speaker diarization, x-vector, probabilistic linear discriminant analysis

1. Introduction

As highlighted in the recent DIHARD 2018 Challenge [1], there are many current techniques for speaker diarization [2, 3, 4, 5, 6, 7]. In this work we begin with systems, like the JHU top-performing system in DIHARD18 [2], that use the established recipe of segmenting speech into approximately fixed-length segments, extracting DNN speaker embeddings [8], length-norm and PCA dimension reduction, clustering with Agglomerative Hierarchical Clustering (AHC) using pairwise PLDA distance [9], and resegmentation with a Variational Bayesian (VB) HMM with eigenvoice priors [10].

While these systems attain good performance, we have two primary concerns. First, they require significant parameter tuning for different environments, as our best configurations for Callhome and DIHARD18 differ in the data used for length normalization (center and whitening) and PLDA training, as well as parameters such as the PLDA dimension, AHC stopping threshold, PCA dimension, VB refinement number of iterations, downsampling, and statistics scaling. Our experience is that suboptimal values of these parameters, for example using Callhome parameters for DIHARD18 evaluation, can easily result in 20% relative performance degradation.

Our second concern is that the power of DNNs is not fully exploited for this task. In particular, DNN embeddings such as x-vectors are trained for general speaker recognition as opposed to diarization, and the resegmentation continues to use generative GMM/factor analysis models without using the DNN.

Therefore our goal in this work is to move towards a unified, DNN-based approach for speaker diarization that maintains state-of-the-art performance across multiple domains without hand-tuning.

2. Combined Clustering and Resegmentation

2.1. Motivation

The VB HMM resgmentation can be viewed as an iterative soft reclustering approach. The downsampling of sufficient statistics present in the algorithm effectively introduces a fixed segmentation (typically about 0.25 seconds), and total factor approximates an i-vector extraction (more precisely a pi-vector [11]). Therefore it is conceptually possible to replace the current two-step process of initial embedding extraction and clustering followed by secondary embedding extraction and refinement with a single process. Note that a VB GMM can also be defined under the PLDA model [12], although this requires the additional assumption that the within-class variability is constant across segments.

After exploring various configurations of joint clustering and variational Bayesian resegmentation using i-vectors, pi-vectors, and x-vectors, we have developed a new leave-one-out PLDA GMM/HMM approach which we will now describe.

2.2. Leave-one-out PLDA GMM/HMM

First, we decide to use the PLDA generative model [13, 14] for segment embeddings. To tackle the problem of joint estimation of speaker models and segment alignments, we propose a leave-one-out (LOO) method to replace the VB approach [12]. Besides giving a practical way to overcome the inherent bias of scoring models against segments which were included in the model estimation, this also allows us to improve performance by removing the independence assumption of PLDA.

In more detail, the LOO PLDA GMM/HMM algorithm works as follows. Given inputs of the length-normalized segment embeddings, initial segment posteriors over speakers, and PLDA parameters (within-class and across-class covariance), alternate between updating the speaker models and generating segment speaker posteriors.

2.2.1. Model Update

As in [15], we use the predictive distribution for each speaker model given the enrollment segments. Given this two Gaussian PLDA model with known covariances $\mathbf{\Sigma}_{wc}$ and $\mathbf{\Sigma}_{ac}$ as well as a set of N enrollment segments $\mathbf{z}_n$, the posterior distribution of the speaker model S_i is Gaussian [16] with mean:

$$\mathbf{m}_i = \mathbf{\Sigma}_{ac} \left(\mathbf{\Sigma}_{ac} + \mathbf{\Sigma}_{ml} \right)^{-1} \bar{\mathbf{z}}_{ml} \tag{1}$$

and covariance:

$$\mathbf{\Sigma}_i = \mathbf{\Sigma}_{ac} \left(\mathbf{\Sigma}_{ac} + \mathbf{\Sigma}_{ml} \right)^{-1} \mathbf{\Sigma}_{ml} \tag{2}$$

where $\bar{\mathbf{z}}_{ml} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{z}_n$ and $\mathbf{\Sigma}_{ml} = \frac{\mathbf{\Sigma}_{wc}}{N}$ represent the maximum likelihood (ML) mean estimate and the covariance of this

http://dx.doi.org/10.21437/Interspeech.2019-2912

estimator. Note that as the number of enrollment segments becomes large, this posterior distribution approaches a delta function at the sample mean. The predictive distribution, is again Gaussian:

$$\mathbf{z}_n|S_i \sim \mathcal{N}(\mathbf{m}_i, \mathbf{\Sigma}_{wc} + \mathbf{\Sigma}_i). \quad (3)$$

In practice, it is well-known that the independence assumptions of PLDA do not work well. In [17], we introduced non-independent enrollment update equations based on a Markov chain assumption (or correlated enrollment). For this model, the enrollment still follows Eq. 1 with the same ML mean, but the covariance of this ML mean estimator now decreases more slowly with increasing number of enrollment segments:

$$\mathbf{\Sigma}_{ml} = \frac{\mathbf{\Sigma}_{wc}}{N}\left(1 + 2\sum_{j=1}^{N-1}\frac{(N-j)}{N}P_r^j\right) \quad (4)$$

The new parameter P_r represents the probability of repeating the channel in successive enrollment segments, or equivalently the correlation between successive channel draws, and allows continuous variation between the two extremes of "by-the-book PLDA scoring" ($P_r = 0$) and "average i-vector scoring" ($P_r = 1$).

To update the speaker priors (weights), we follow [18, 10] and use a non-Bayesian maximum-likelihood approach, as it has been observed to have good properties of eliminating redundant speakers.

2.2.2. Speaker Assignments

The updated speaker models (means, covariances, and weights) form a Gaussian mixture model, so speaker assignment is done by computing LOO posteriors per class (responsibilities). Optionally, we add a simple HMM to discourage rapid speaker transitions using the approach of [10], where a single parameter P_{loop} controls the probability that the speaker is the same as the previous segment.

2.2.3. Dimension Reduction and Diagonalization

To reduce computation, this work uses diagonal covariance matrices based on the fact that two symmetric matrices can be simultaneously diagonalized with a linear transformation [19]. This process is similar to Linear Discriminant Analysis (LDA) and is given by:

1. perform eigendecomposition $\mathbf{\Sigma}_{wc} = E_1\Lambda_1 E_1^T$
2. transform $\mathbf{\Sigma}_{ac}$ with $\mathbf{\Sigma}'_m = \Lambda_1^{-\frac{1}{2}} E_1^T \mathbf{\Sigma}_{ac} E_1 \Lambda_1^{-\frac{1}{2}}$
3. perform eigendecomposition $\mathbf{\Sigma}'_m = E_2\Lambda_2 E_2^T$
4. (optional) keep only principal components
5. final transform: $\mathbf{z}'_n = E_2^T \Lambda_1^{-\frac{1}{2}} E_1^T \mathbf{z}_n$

In this transformed space, $\mathbf{\Sigma}_{wc} = I$ and $\mathbf{\Sigma}_{ac} = \Lambda_2$.

2.3. Selecting Number of Speakers

In all clustering applications, selecting the number of clusters is a challenging problem. While AHC with PLDA comparisons does work well for this task, it works best with a tuned, task-dependent stopping threshold. As in [10], we prefer to start with a maximum number of speakers and let the clustering algorithm automatically select the correct number. While the LOO PLDA GMM/HMM does produce overall likelihood estimates for any number of speakers, in practice the ML weight updates quickly reach zero for unnecessary speakers. We initialize the algorithm with k-means in the diagonalized embedding space with a fixed number of speakers, and let the algorithm eliminate speaker weights across iterations.

Table 1: *Baseline x-vector architecture. The full model has layer size L and embedding dimension D set to 512, but these experiments use a smaller model with both set to 256.*

Layer	Layer Type	Context	Size
1	TDNN-ReLU	t-2:t+2	L
2	Dense-ReLU	t	L
3	TDNN-ReLU	t-2, t, t+2	L
4	Dense-ReLU	t	L
5	TDNN-ReLU	t-3, t, t+3	L
6	Dense-ReLU	t	L
7	TDNN-ReLU	t-4, t, t+4	L
8	Dense-ReLU	t	L
9	Dense-ReLU	t	3*L
10	Pooling (mean+stddev)	Full-seq	6*L
11	Dense(Embedding)-ReLU		D
12	Dense-Softmax		Num. spks.

3. DNN Training

We would like to train the DNN embedding directly for diarization, i.e. to optimize the performance of the LOO PLDA GMM/HMM algorithm. In this section we demonstrate progress in this direction.

3.1. Gaussian Linear Layer

The baseline DNN embedding is a simplified version of the latest x-vector architecture [20] as shown in Table 1. Our first step is to modify this DNN architecture to better generate Gaussian embeddings. It is well-known that a multiclass Gaussian classifier with shared covariance leads to a linear classifier, which can be implemented with DNN embeddings which drive a linear layer into a softmax [21]. Therefore our first modification to the architecture is to convert the last nonlinear layer 11 into a linear bottleneck by removing the final ReLU non-linearity and also constrain the bias of the last layer to depend on the means [21]. We refer to this as a discriminative Gaussian linear layer.

3.2. Generative Gaussian Linear Layer

We now shift towards a generative model by removing the model means from DNN training and replacing them with maximum likelihood estimates. While these could be within a mini-batch as in generalized end-to-end cost (GE2E) [22], we propose here to instead use past values of embeddings for the class. In particular, our generative Gaussian linear last layer updates each model with the mean of the last N embeddings for that class.

3.3. Length-norm Layer

Length normalization is well-known to improve the performance of Gaussian PLDA models [23], so we add an additional layer after the embedding for this. We have experimented with explicitly modeling the centering and a whitening matrix, or adding penalties to force DNN training to model them, but in practice we find that we don't need to add these parameters as the DNN can directly drive the embeddings to make length-normalization work.

3.4. PLDA and Generative Gaussian Quadratic Layer

To implement PLDA scoring in the DNN, we need to perform Bayesian model enrollment using Eq. 1. The Gaussian quadratic outer layer maintains enrollment statistics as in the generative Gaussian linear layer, and then combines these statistics with the PLDA parameters $\mathbf{\Sigma}_{wc}$, $\mathbf{\Sigma}_{ac}$, and $\mathbf{U}_{lda}$ to produce predictive distributions for each class. Finally, log-likelihoods for each class are produced with Eq. 3.

The PLDA parameters are estimated in the following way. First, the within-class covariance $\mathbf{\Sigma}_{wc}$ is set to identity, as we again assume the DNN can force the embeddings to match this property. We do need a discriminatively-trained scale factor to compensate for the length-normalization constraint. For the across-class covariance $\mathbf{\Sigma}_{ac}$, we fear that discriminative training will encourage overtraining, as the DNN could overcome the Bayesian uncertainty for the known training data. Therefore we use a generative update of this matrix. The PLDA layer maintains it's own separate copy of enrollment statistics over previous embeddings for each class, and $\mathbf{\Sigma}_{ac}$ is approximated by the covariance of these ML model means. Given these two covariances, the diagonalizing transform is periodically updated with the eigendecomposition described above. We start with full-rank matrices, but find that the PLDA layer automatically learns the dimension of speaker space as the rank of the across-class covariance reduces over time.

3.5. Gaussian Diarization Cost

We desire an error metric that will predict performance of LOO PLDA GMM/HMM diarization. For each minibatch, we create a synthetic diarization task with the following generative model:

- Draw a random set of M speakers.
- For each speaker, draw a random number of segments.

Rather than implement the full clustering algorithm within DNN training, we instead require the network to optimize the posteriors of each segment given the true speaker labels, as non-divergence from the true answer is closely related to clustering convergence. The posteriors for each speaker are computed using the oracle speaker weight (from the labels) and LOO PLDA enrollment from the remaining speaker segments within the minibatch; the diarization cost is defined as the normalized cross-entropy between the truth labels and these posteriors. Note that this diarization cost can be viewed as a probabilistic extension of GE2E [22].

As shown below, this synthetic diarization cost corresponds quite well to an actual diarization error rate. Since it is computed directly from embeddings and PLDA parameters without any outer layer, meaningful validation can be computed during DNN training using held-out speakers. In addition, this cost can also be used directly to train the embeddings.

4. Experimental Results

We train with LDC corpora Switchboard, Fisher, Mixer6, SRE2004-10, and VoxCeleb1 [24]. With augmentations, this results in a set of 456235 utterances from 14846 speakers. We use a 90/10 split between training and development sets, so the training set uses 13361 speakers, with minibatches of size 128 with one segment per speaker. Segments vary between 1.5 and 2.5 seconds. The synthetic diarization cost used a held-out set of 1485 speakers, with minibatches of 640 segments from 64 speakers with random number of segments between 2 and 20 per speaker. Both training and validation cost are normalized cross-entropy (NCE), where normalizing by the logarithm of the number of classes facilitates comparison across disparate class sizes. NCE should range between 0 (perfect) and 1 (random), although miscalibrated systems can exceed 1.

Table 2: *DNN training performance. Train error is presented as NCE, and DiarCost is validation synthetic diarization cost.*

System	Train NCE	DiarCost	PLDAdim
Baseline x-vector	0.13	-	256
GaussDiscr	0.12	12.0	256
Gauss100ml	0.15	0.32	256
Gauss100Bayes	0.18	0.08	123
Gauss10Bayes	0.22	0.08	93
Gauss1Bayes	0.63	0.11	67
Gauss1-20Bayes	0.34	0.07	97
DiarCost	0.09	0.09	120

4.1. DNN Training Performance

As shown in Table 2, this baseline x-vector system achieves quite good training error ($NCE = 0.12$); synthetic diarization cost cannot be measured since this network has no internal PLDA. The discriminative Gaussian system (GaussDiscr) performs similarly during training, and has internal PLDA parameters. Unfortunately, since this DNN does not use PLDA to compute training cost, diarization cost is much worse than random. A generative Gaussian linear DNN using the previous 100 speaker segments for maximum likelihood enrollment (Gauss100ml) is 10-20% worse at training, but has diarization cost much less than one. During training, this network is aware of the need to drive the within-class covariance toward identity but unaware of the across-class covariance.

The introduction of Bayesian enrollment and scoring via the quadratic Gaussian layer (Gauss100Bayes) makes all of the PLDA parameters relevant during training. This further increases training error, but significantly decreases diarization cost. Note that in all systems the across-class covariance is updated using the previous 30 segments per class. As we reduce the number of previous segments for Bayesian enrollment to 10 (Gauss10Bayes) and then 1 (Gauss1Bayes), the training error continues to increase while diarization cost stays similar. Note that the single enrollment system struggles to attain even mediocre training performance, as the task of speaker comparison from single 2 second segments is very difficult, but it does maintain a reasonable convergence behavior. We did not expect that such a challenged system could learn both embeddings and PLDA parameters from a random start. The PLDA dimensions learned by these networks are all much less than full rank of 256, and decrease further as the training task becomes harder with less enrollment.

The best diarization cost is achieved by Bayesian enrollment from a random number of previous segments (Gauss1-20Bayes). Optimizing diarization cost (DiarCost) directly is competitive with this system but not as good. We believe this is due to more limited data shuffling (multiple segments per speaker in a minibatch) and fewer competitors in the cross-entropy metric (64 vs. 13,361).

4.2. Diarization Experiments

For the NIST SRE2000 Callhome diarization task, we follow standard practice and report results for oracle speech activity

Table 3: *Callhome Diarization Error Rate (DER).*

System	Parameters	DER
PLDA AHC+VB [25]	task-dependent	9.9
VB HMM [10]	task-dependent	9.0
UIS-RNN [26]	general	8.2
UIS-RNN [26]	task-dependent	7.6
Baseline x-vector	task-dependent	8.9
GaussDiscr	task-dependent	7.7
Gauss100ml	task-dependent	8.4
Gauss100ml	general	34.1
Gauss100Bayes	general	10.9
Gauss10Bayes	general	7.5
Gauss1Bayes	general	11.0
Gauss1-20Bayes	general	**7.1**
DiarCost	general	8.4

Table 4: *Performance vs. enrollment repeat probability (segment correlation).*

Train	Test	Train NCE	DiarCost	DER
0.0	0.0	0.32	0.074	9.3
-	0.9	-	-	12.0
0.9	0.0	-	-	16.5
-	0.9	0.34	0.069	7.1
-	1.0	-	-	11.7
1.0	0.9	-	-	12.0
-	1.0	0.35	0.074	8.6

marks with 250 ms forgiveness collar around speaker change points. The first four rows in Table 3 present the best published results for this task. The column labeled "Parameters" represents whether the system parameters are explicitly trained for the Callhome task or internal to the general DNN model.

The second part of this table shows results with our LOO PLDA GMM system. The systems are the same as presented in the previous table. With the x-vector baseline, parameters such as centering, whitening, and PLDA parameters are externally trained. The newer DNN systems can be run in this way also; however they typically perform better with the general model parameters generated during DNN training. With these internal parameters, there is no system adjustment for the Callhome task; diarization is simply run straight from the DNN model itself. In comparing these numbers to the DNN training performance of Table 2, we see that training NCE has little relation to actual DER performance, while synthetic diarization cost is quite a good predictor. We also see that, as in the previous table, our best LOO PLDA DNN is the one trained with Bayesian enrollment with a variable number of segments, so this is the system we use in the contrastive experiments below. This out-of-the-box performance is better than any published number for this task, and competitive with our best internal PLDA AHC+VB systems using x-vectors [8].

4.3. Algorithm Permutations

We trained DNNs with three different values of non-independent enrollment, with $P_r = 0$ (by-the-book PLDA), $P_r = 0.9$, and $P_r = 1$ (average i-vector scoring). Table 4 confirms that the dependent model works best, but training the DNN to generate independent embeddings is at least partially successful. Note that $P_r = 1$ has the same uncertainty for every model, and is very similar to non-probabilistic cosine scoring.

To attain better timing refinement, we would like to use shorter segments but this makes clustering more difficult. As shown in Table 5, one second segments require an HMM for good performance, but still fall short of the 2 second version.

Table 5: *Segment length performance comparison.*

System	Train NCE	DiarCost	DER
2 sec GMM	0.32	0.07	7.1
2 sec HMM	-	-	**7.0**
1 sec GMM	0.51	0.15	8.5
1 sec HMM	-	-	7.5

Table 6: *DIHARD18 development set performance.*

System	DER
PLDA AHC x-vector [2]	23.4
PLDA AHC wideband x-vector + VB HMM [2]	18.2
Gauss1-20Bayes HMM	19.9
Diarcost HMM	18.9

4.4. Robustness

Our best systems reported above achieve state-of-the-art DER for Callhome without any training or tuning to that task. To further test whether these systems meet our initial objective of robustness without tuning, we now test them on the completely different task of the DIHARD 2018 challenge [2]. We report results on the DIHARD18 development set, which covers ten diverse domains including monologues, interviews with children, meetings, and internet videos. We report results for Track 1, using the oracle SAD marks but no collars in measuring DER.

Table 6 compares two of our new systems against published results from the JHU winning system for this challenge task. The first baseline is a PLDA AHC x-vector system tuned for this task. The second baseline is a PLDA AHC + VB HMM system based on x-vectors which exploit the wider bandwidth available in these microphone recordings. Both of our new DNN systems are significantly better than this first baseline and almost as good as this top-performing wideband system, despite being constrained to narrow bandwidth for compatibility across tasks. These represent JHU's best results to date on this task with a narrowband system design.

5. Conclusion

This paper has presented two contributions: an improved HMM segment assignment algorithm using leave-one-out Gaussian PLDA scoring, and an approach to training the DNN such that embeddings directly optimize performance of this scoring method with generatively updated PLDA parameters. Initial experiments with this new system are very promising, achieving state-of-the-art performance for two separate tasks (Callhome and DIHARD18) without task-dependent parameter estimation or tuning.

6. References

[1] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "First DIHARD challenge evaluation plan," 2018.

[2] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe *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.

[3] M. Diez, F. Landini, L. Burget, J. Rohdin, A. Silnova, K. Zmolıková, O. Novotnỳ, K. Veselỳ, O. Glembek, O. Plchot *et al.*, "But system for DIHARD speech diarization challenge 2018," in *Proc. Interspeech*, 2018, pp. 2798–2802.

[4] L. Sun, J. Du, C. Jiang, X. Zhang, S. He, B. Yin, and C.-H. Lee, "Speaker diarization with enhancing speech for the first DIHARD challenge," *Proc. Interspeech 2018*, pp. 2793–2797, 2018.

[5] Z. Zajıc, M. Kunešová, J. Zelinka, and M. Hrúz, "Zcu-ntis speaker diarization system for the DIHARD 2018 challenge," in *Proc. INTERSPEECH*, 2018, pp. 2788–2792.

[6] I. Vinals, P. Gimeno, A. Ortega, A. Miguel, and E. Lleida, "Estimation of the number of speakers with variational bayesian plda in the DIHARD diarization challenge," in *Proc. INTERSPEECH*, 2018, pp. 2803–2807.

[7] J. Patino, H. Delgado, and N. Evans, "The eurecom submission to the first DIHARD challenge," in *Proc. Interspeech*, vol. 2018, 2018, pp. 2813–2817.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[9] G. Sell and D. Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 413–417.

[10] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on bayesian hmm with eigenvoice priors," in *Proceedings of Odyssey*, 2018, pp. 147–154.

[11] D. Garcia-Romero and A. McCree, "Subspace-constrained supervector plda for speaker verification." in *Interspeech*, 2013, pp. 2479–2483.

[12] J. Villalba, A. Ortega, A. Miguel, and E. Lleida, "Variational bayesian plda for speaker diarization in the mgb challenge," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 667–674.

[13] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[14] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. ICCV*, 2007, pp. 1–8.

[15] B. J. Borgstrom and A. McCree, "Discriminatively trained Bayesian speaker comparison of i-vectors," in *Proc. ICASSP*, 2013.

[16] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley, 2001.

[17] A. McCree, G. Sell, and D. Garcia-Romero, "Extended variability modeling and unsupervised adaptation for plda speaker recognition." in *Interspeech*, 2017, pp. 1552–1556.

[18] P. Kenny, "Bayesian analysis of speaker diarization with eigenvoice priors," *CRIM, Montreal, Technical Report*, 2008.

[19] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[20] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[21] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing*. Springer, 1990, pp. 227–236.

[22] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[23] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. Interspeech*, 2011, pp. 249–252.

[24] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[25] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4930–4934.

[26] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," *arXiv preprint arXiv:1810.04719*, 2018.