The Speaker and Language Recognition Workshop (Odyssey 2022)
28 June - 1 July 2022, Beijing, China

Speaker recognition on mono-channel telephony recordings

Yosef Solewicz[1], *Noa Cohen*[1], *Johan Rohdin*[2], *Srikanth Madikeri*[3], *Jan "Honza" Čercnocký*[2]

[1]Ministry of Public Security – Israel National Police, Israel
[2]Brno University of technology, Czechia
[3]IDIAP research institute, Switzerland

yosef.solewicz@gmail.com, noa11120@walla.co.il
{rohdin, cernocky}@fit.vutbr.cz, srikanth.madikeri@idiap.ch

Abstract

Conversations stored as mono data is a common problem in many real world speaker recognition applications. In this paper, we focus on investigative scenarios, where a number of mono telephone conversations are available for a speaker of interest. For example, a human operator may have verified that the speaker is present in these conversations. We propose several approaches for automatically creating enrollment models for the speaker of interest from such data. We then use the enrollment models to search for appearances of the speaker of interest in other calls. We analyze the performance of the different method on two dataset that matches our scenario, one is from a simulated case and one is from a real case.

1. Introduction

In many practical applications of speaker verification, the recordings for both enrollment and verification contain the speech from more than one speaker. This problem has been adressed in several speaker recognition evaluations, e.g. Speakers In The Wild (SITW) [1] and NIST SRE18 [2]. Our interest in this paper is telephony data where the two sides of the call have been mixed and stored as one mono recording. In our scenario, we need to create enrollment models from mono recordings as well as verify whether a specific speaker is present in a test utterance. This scenario is common in criminal investigations involving lawfully intercepted telephone calls.

Assuming that an enrollment model exists, a common and straight-forward approach to do verification, used e.g. in [3], is to first apply speaker diarization to the test utterance and then score each of the obtained speaker embeddings found by the diarization against the enrollment model. As the final verification score, the maximum of the obtained scores is used. This simple approach works well and has sound mathematical justification as will be explained later in this paper.

Creating the enrollment model is a much more difficult problem. In SITW [3], multiple speakers exist in the enrollment recording, and a segment where the speaker to enroll is speaking is marked (referred to as the *assist* segment). The rest of the enrollment recording was diarized, all resulting speakers were scored against the assist segment. The best scoring segment was used in addition to the assist segment to create the enrollment model.

In our scenario, no *assist* segments are available. However, we do have several calls for the enrollment speaker where it can be assumed that the conversation partners are different in the different calls. In this paper, we compare different methods to create speaker models from such data. It should be noted that in our scenario, it is important to create a model for the speaker of interest while excluding the conversation partners in the enrollment calls because some of them may occur in the test recording speaking with someone else than the enrollment speaker and therefore generate a false positive. We cannot simply diarize/cluster the enrollment recordings and score every obtained model from the enrollment data with the test recording to verify whether at least one speaker in the enrollment recordings is the same as one speaker in the test recordings.

To the best of our knowledge, the problem of enrolling specific speakers from several multi-speaker recordings is seldom addressed. However, this type of data has been utilized for training a neural network based *speaker identification* system in [4]. In that work, utterances where the identity of at least one speaker was known were diarized and segmented to chunks from which i-vectors [5] were extracted. The speaker identification system was trained using a novel objective function that took into account the properties of the data.

The paper is organized as follows: in Section 2, we discuss diarization approaches used in this work, especially explaining how to constrain the number of detected speakers which is important in our scenario. In Section 3, we describe the proposed approaches for creating enrollment models. In Section 4, we suggest the verification procedure and justify it. In Section 5, we present experiments done on two datasets. Finally, in Section 6 we summarize our conclusions and outline some directions for future work.

2. Diarization

In this work, we use diarization in the verification phase as well as for for some of the methods for creating enrollment models. Since we are dealing with telephony recordings, we assume that there are two speakers in the recordings[1]. We explore two diarization approaches in particular that achieves this. These are described in the following two subsections.

2.1. PCA based diarization

A simple approach to diarization of two speakers was introduced in [6, 7, 8]. This method implements an implicit seg-

[1] Of course this is not always the case. There can be three speakers if one of the phones is handed over to another person in the middle of the call, or there can be only one speaker if someone makes a call only to give a short instruction. However, having two speakers in the recording is by far the most common and in this work we restrict ourselves to this case.

10.21437/Odyssey.2022-27

mentation (at the vector level), fitted for a subsequent verification step, since no explicit diarization at the speech signal is performed. It was originally applied to GMM supervectors but it can be naturally applied to any type of speaker embedding. The embeddings used in this paper are x-vectors (DNN based speaker embeddings) produced from short speech chunks (1.44s in our case) with some overlap (0.24s in our case). These embeddings are then centered and projected into their first principal component (the eigenvector corresponding to the largest eigenvalue) and those that obtain a value larger than t are assigned to one speaker, and to the other speaker otherwise. Here t is a tunable threshold which can be the average of the conversation projection scores or simply set to 0, as in our experiments. Furthermore, some floor threshold could be used to eliminate borderline (possibly unreliable) projected x-vectors. Intuitively, the main eigenvector retains the direction of maximum embedding variability which is associated to the speaker separability, while more refined phonetic separability would be manifested along higher order eigenvectors. Note that this diarization method results in exactly two speakers.

2.2. VBx diarization

Variational Bayes diarization on top of speaker embeddings (VBx) is a state-of-the- art method for diarization described in detail in [9]. In short, it is a first order hidden Markov model for transitions between speakers where the output probabilities are modeled by probabilistic linear discriminant analysis (PLDA) [10]. As the method described in the previous subsection, VBx takes speaker x-vector embeddings from short segments of an utterance (hence the *x* in *VBx* since it was first applied to x-vectors). Usually, and in this paper, agglomerative hierarchical clustering (AHC) is applied for initializing the method.

The standard VBx recipe does not provide a mechanism for constraining the number of detected speakers. Given a *minimum*, and *maximum* on the number of speakers, we therefore modify the recipe as follows:

1. The AHC used for intitialization is stopped if the minimum number of speakers is reached. As usual, it is also stopped if a threshold in the similarity score for the two most similar clusters is reached.
2. The VB diarization then outputs the best solution that does not have less than the minimum number of required speakers.
3. In a post-processing step, if there are more than the *maximum* number of required speakers, the speakers with most segments are selected and the remaining segments are reassigned to one of these speaker based on the posterior probability of them belonging to the different speakers.

Note that AHC always reduces the number of speakers in every iteration and that the VB diarization cannot increase the number of speakers to more than what is available in its initialization. In our experiments, both the minimum and maximum number of speakers is set to two which means that exactly two speakers will be obtained.

3. Methods for enrollment

In this section, we describe three methods for automatically building a target speaker model given some multi-speaker conversations where the target speaker occurs. In particular, we focus on mono (two-wire) telephone conversations. Assume we have n mono conversations where speaker S was identified, for example by a human operator or by the telephone number. Supposing that S speaks to different partners in each of the n conversations, the conversations would contain $n+1$ clusters corresponding to each of the different speakers. An obvious solution for modeling S would be to manually label the conversations and isolate his/her speech chunks to derive the model. This procedure leads to an optimal speaker model but at high time and effort costs. Our goal is to limit human workload by proposing alternative techniques to automatically produce approximate models.

We propose three methods for automatic speaker enrollment from mono-channel calls, "**median** embedding", "eigenvector **intersection**" and "Complete **cluster** search", where the bold denotes how we refer to them in figures and tables. The proposed methods are described in detail in the subsections below. An illustration of the methods is provided in Figure 1. It shows a 2D PCA visualization of segmental embeddings extracted from three conversations, each involving the target speaker and one other speaker (red, green and blue dots for each of the three conversations). The overall median, the intersection and the top cluster embedding are plotted as different (yellow) symbols. For completeness, each of the $(2n)$ initial clusters and the conversations eigenvectors (black dots) are also plotted. Note that the intersection and cluster methods produced similar speaker models, in contrast to the median, which was biased by the relatively high amount of x-vectors from of the other speaker in the "green" conversation.

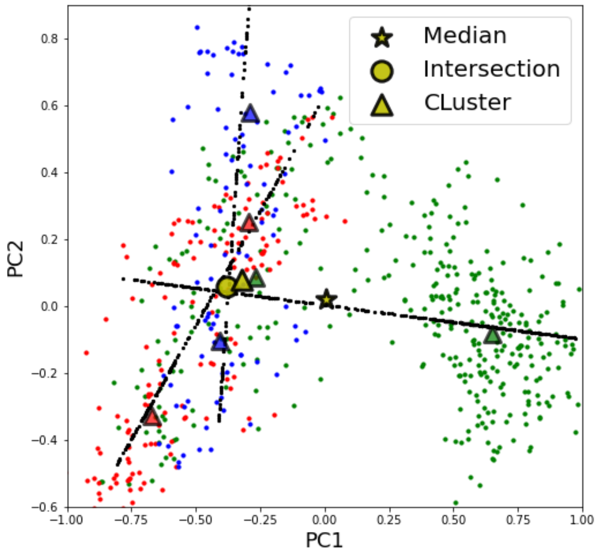

Figure 1: Illustration of the Median, Intersection and Cluster methods for three calls. The dots represents the embeddings from the short segments described in Section 2. Each color denotes a call. For illustrative purposes, the data has been projected on the two first principal components (PC) estimated on all data. (Distinct color Cluster symbols refer to the centers obtained by PCA diarization for different conversations.)

3.1. Median embedding

Possibly the most trivial model for speaker S is simply the element-wise median (or mean) embedding calculated from the whole set of embeddings extracted from his/her n enrollment conversations. The rational behind this approach is that the "center of mass" of this set should be representative of the feature space region concentrating most of the embeddings of S, while the embeddings from the (at most) n partners should be spread over different directions of the feature space. In our experiments, we use the median rather than the mean because, as opposed to the mean, the median is less influenced by the embeddings of the partner speakers (since the target speaker is present in all enrollment calls and therefore has many more embeddings than the conversation partners). In fact, this method simultaneously operates at the whole enrollment data for a certain speaker and not on his individual conversations. More sophisticated strategies separately processing each enrollment conversation will be presented in the next subsections.

3.2. Eigenvectors intersection

Inspired by the previously described PCA diarization, we assume that the main eigenvector "connects" the partner speakers embeddings in a conversation. Therefore, given n conversations of speaker S, the intersection of their eigenvectors would point to the concentration of his embeddings, since S is the common speaker in each of the conversations. We can then approximate the speaker's model by $\boldsymbol{p}$, the closest point from the eigenvectors intersection, which can be found through the following relation:

$$\sum_i \left[\boldsymbol{n}_i \boldsymbol{n}_i^T - \boldsymbol{I}\right] \boldsymbol{p} = \sum_i \left[\boldsymbol{n}_i \boldsymbol{n}_i^T - \boldsymbol{I}\right] \boldsymbol{a}_i, \qquad (1)$$

where I is the identity matrix, $\boldsymbol{a}_i$ is the mean embedding for conversation i and $\boldsymbol{n}_i$ is the normalized main eigenvector of this conversation. The eigenvectors most certainly won't have a unique intersection and the pseudo-inverse should be used to calculate the "best fit" (in the least squares sense) solution to $\boldsymbol{p}$. (The formula derivation for general line-to-line intersections can be found in [11].)

3.3. Complete cluster search

In this method we first diarize the enrollment calls so that we obtain two speaker embeddings per call. We take for granted that one out of the two clusters formed for each enrollment conversation belongs to the speaker of interest. For n enrollment recordings there are therefore 2^n ways to combine the embeddings belonging to the target speaker. Each of the 2^n combinations is then evaluated according to some objective and the best one is selected. In this work we explore two objectives. The first is simply the average standard deviation of the embeddings. The second objective is the likelihood according to the PLDA model. Let $c \in [1, 2^n]$ denote a combination of one embedding per utterance. If c is the correct combination, these embeddings belong to the target speaker. Let them be denoted $\boldsymbol{E}^{(c)} = [\boldsymbol{e}_1^{(c)}, \dots \boldsymbol{e}_n^{(c)}]$. The remaining embeddings are assume to belong to n different speakers. Let them be denoted $\boldsymbol{E}^{(\bar{c})} = [\boldsymbol{e}_1^{(\bar{c})}, \dots \boldsymbol{e}_n^{(\bar{c})}]$. The likelihood for cluster c is then

$$L(c) = P(\boldsymbol{e}_1^{(c)}, \dots, \boldsymbol{e}_n^{(c)}) \prod_{i=1}^{n} P(\boldsymbol{e}_i^{(\bar{c})}), \qquad (2)$$

where $P(\boldsymbol{x}_1, \dots, \boldsymbol{x}_m)$ denotes the probability of observing the embeddings $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$ if they are from the same speaker and is given by Eq. (6) in [10]. If all combinations have equal prior probability, their posterior probabilities can be obtained by applying a softmax over all likelihoods.

It should be noted that this approach is only feasible as long as n is reasonably small. For example, with $n = 20$ there are approximately 10^6 combinations. In this case it takes 1 to 2 minutes to search for the best combination and increasing n by 1 doubles the processing time. On the other hand, it is unlikely that using more than 20 calls is needed for creating a good speaker model.

4. Verification

As discussed earlier, in this work we assume there will always be two speakers in the call. Using the methods described in Section 2, we diarize the utterance and obtain an average embedding for each of the speakers (A,B) in the test call.

Given these test utterance embeddings are obtained, a standard approach for verification used e.g. in [3] is to score the enrollment embedding against each test embedding and use the maximum score as the verification score. This process is illustrated in Figure 2.

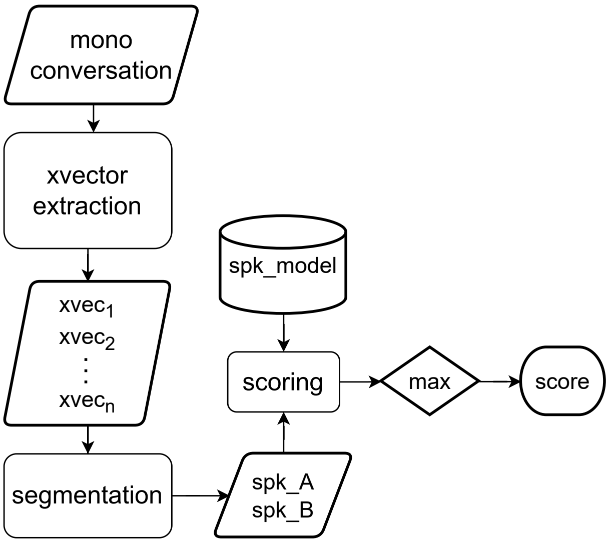

Figure 2: General verification scheme. In the diarization stage, the mono conversation to test is divided into short chunks from which x-vectors are extracted. These segmental x-vectors are then clustered (segmentation) into two speakers and one x-vector for each speaker is obtained by averaging. In scoring, each of the two x-vector is compared against the speaker model and the maximum score is used as the final score.

Let t_A be the embedding from speaker A in the test recording and t_B be the embedding from speaker B in the test recording. Let $T = \{t_A, t_B\}$. Further, let $E = \{e_1, \dots, e_n\}$ be the enrollment embeddings for the speaker of interest, i.e., one per enrollment utterance.

In the following, we show that using the maximum of the two scores can be seen as an approximation to the LLR of between the following two hypothesis

- $\mathcal{H}_1$: One of the embeddings (t_A, t_B) from the test utterance is from the same speaker as the enrollment embeddings
- $\mathcal{H}_0$: None of the embeddings from the test utterance is from the same speaker as the enrollment embeddings.

These two hypothesis are the only possible given the assumption that the system have detected the correct number of speakers.

The LR is

$$LR = \frac{P(E,T|\mathcal{H}_1)}{P(E,T|\mathcal{H}_0)} = \frac{P(\mathcal{H}_1|E,T)}{P(\mathcal{H}_0|E,T)} \frac{P(\mathcal{H}_0)}{P(\mathcal{H}_1)} \tag{3}$$

To derive the first term, we consider for simplicity the case of two test embeddings t_A and t_B. We then divide $\mathcal{H}_1$ into

- $\mathcal{H}_{11}$: The speaker in embedding t_A is from the same speaker as the enrollment embeddings
- $\mathcal{H}_{12}$: The speaker in embedding t_B is from the same speaker as the enrollment embeddings.

We then have

$$\begin{aligned}\frac{P(\mathcal{H}_1|E,t_A,t_B)}{P(\mathcal{H}_0|E,t_A,t_B)} &= \frac{P(\mathcal{H}_{11}|E,t_A,t_B) + P(\mathcal{H}_{12}|E,t_A,t_B)}{P(\mathcal{H}_0|E,t_A,t_B)} \\ &= \frac{P(E,t_A,t_B|\mathcal{H}_{11})P(\mathcal{H}_{11})}{P(E,t_A,t_B|\mathcal{H}_0)P(\mathcal{H}_0)} + \frac{P(E,t_A,t_B|\mathcal{H}_{12})P(\mathcal{H}_{12})}{P(E,t_A,t_B|\mathcal{H}_0)P(\mathcal{H}_0)} \\ &= \frac{P(E,t_A|\mathcal{H}_{11})P(\mathcal{H}_{11})}{P(E,t_A|\mathcal{H}_0)P(\mathcal{H}_0)} + \frac{P(E,t_B|\mathcal{H}_{12})P(\mathcal{H}_{12})}{P(E,t_B|\mathcal{H}_0)P(\mathcal{H}_0)} \end{aligned} \tag{4}$$

where the last simplification comes because e.g., $P(E,t_A,t_B|\mathcal{H}_{11}) = P(t_B)P(E,t_A|\mathcal{H}_{11})$ and similarly for other parts. Without more detailed information about the recording, each embedding is equally likely to be the one from the enrollment speaker so p_{tar} is therefore distributed uniformly over the test embedding, i.e., $P(\mathcal{H}_{11}) = P(\mathcal{H}_{12}) = P(\mathcal{H}_1)/2$. This, together with Eq. (3) and Eq. (4), gives the LR:

$$\begin{aligned} LR &= \frac{P(E,T|\mathcal{H}_1)}{P(E,T|\mathcal{H}_0)} \\ &= \frac{1}{2}\left[\frac{P(E,t_A|\mathcal{H}_{11})}{P(E,t_A|\mathcal{H}_0)} + \frac{P(E,t_B|\mathcal{H}_{12})}{P(E,t_B|\mathcal{H}_0)}\right], \end{aligned} \tag{5}$$

i.e., it is simply the sum of the LR for the individual test embeddings. Accordingly, the LLR can then be obtained by

$$\begin{aligned} LLR &= \log\left[\frac{P(E,t_A|\mathcal{H}_{11})}{P(E,t_A|\mathcal{H}_0)} + \frac{P(E,t_B|\mathcal{H}_{12})}{P(E,t_B|\mathcal{H}_0)}\right] - \log(2) \\ &= \log\left(\exp s\left(t_A,E\right) + \exp s\left(t_B,E\right)\right) - \log(2), \end{aligned} \tag{6}$$

where

$$s(t,E) = \log\frac{P(E,t)}{P(E)P(t)} \tag{7}$$

is the ”standard” LLR score. Except for the constant term, this can be approximated with the max operation. In initial experiments we observed no difference (for calibration insensitive evaluation metrics) between using the maximum or the formula in Eq. (6) so we use the maximum for simplicity.

5. Experiments

In this section, we first introduce the backbone models (embedding extractor and PLDA) used in the experiments. We then present experiments on one simulated and one real dataset from a closed and completed criminal investigation.

5.1. Models

In all experiments we used the $8kHz$ embedding extractor and PLDA model used and described in [9]. This model is publicly available as part of the VBx recipe[2]. In short, the embedding extractor is a ResNet101 architecture, extracting 256 dimensional embeddings. Before PLDA, the embeddings are subjected to a rather involved preprocessing step as follows,

$$\boldsymbol{x} = \left\| \| \boldsymbol{x}_o - \boldsymbol{\mu}_1 \|_2 \mathbf{D} - \boldsymbol{\mu}_2 \right\|_2 \tag{8}$$

where $\|\cdot\|_2$ is the $L2$-norm, $\boldsymbol{x}_o$ is the original embedding of dimension 1×256, $\mathbf{D}$ is an 256×128 dimensional LDA projection matrix, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are row vectors, and $\boldsymbol{x}$ is the preprocessed embedding. The paramameters $\mathbf{D}$, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are estimated from the PLDA training data. The PLDA model has full (rank 128) within and between class covariance matrix.

5.1.1. A note on PLDA scoring

In most of our experiments, we use the log-likelihood ratio from a PLDA model for scoring which can be computed in closed form. The formulas for this is given in, e.g., [10]. The PLDA LLR computation is based on the number of enrollment utterances, the mean of the embeddings from the enrollment utterances and the embedding from the test utterance. The median and intersection method do not, strictly speaking, produce a mean vector and it is somewhat unclear what should be considered the number of enrollment utterances. Further, since the preprocessing step described in Eq (8) is non-affine, it makes a difference if it is applied before or after ”merging” embeddings. For the median and intersection method we therefore produce the enrollment embedding, then apply the PLDA preprocessing and then in scoring we set the number of enrollment utterances to 1. For the complete cluster search, we obtain a set of enrollment embeddings. Therefore we can also apply the preprocessing before merging them by taking their mean. In the experiments we try both ways as well as setting the number of enrollment utterances to 1 (referred to as ”mean scoring”) and the actual number (referred to as ”by-the-book” scoring).

5.1.2. Evaluation metric

For a given enrollment model and test call, the task is to tell whether the speaker of the enrollment model is present in the test call regardless of which side of the test call he/she is in. This is a detection task. For simplicity, we use equal error rate (EER) as evaluation metric.

5.2. Roxanne simulated data

The Roxanne simulated data set (ROXSD) data set [12] was collected within the Roxanne project[3] to aid the development of speech, NLP, video and network analysis tools for crimininal investigations[4]. It contains partially scripted calls between member of a three connected criminal networks. The scenarios are inspired by real criminal cases. In the ROXSD scenario, ten

[2] `https://github.com/BUTSpeechFIT/VBx`

[3] Real time network, text, and speaker analytics for combating organized crime (Roxanne), `https://roxanne-euproject.org`

[4] The authors do not see any significant ethical or privacy concerns that would prevent the processing of the data used in the study. These data were collected from consenting volunteers, whose participation was approved by a research ethics committee. The dataset do contain personal data, and these are processed in compliance with the GDPR and national law.

Table 1: Statistics for ROXSD data. Two of the intercepted speakers had only one intercepted call and were not used. Note that the sum of the last two rows equals $236 - 8 = 228$ for each speaker.

Speaker ID	#Target calls	#Non-target calls
G03_M	-	-
V02_M	-	-
C04_M	27	201
R06_F	6	222
C07_F	37	191
G01_M	27	201
R05_M	7	221
R01_M	25	203
V01_M	3	225
C01_M	48	180
Sum	180	1644

volunteers playing the role of fictitious criminals having their phone numbers intercepted. There are in total 236 calls in the data set. We use the eight chronologically first calls of each intercepted speaker to create enrollment models. The rest of their calls are used as target calls. The non-target calls for each speaker are taken from the rest of the speakers in the data including calls that are used to create enrollment models for other speakers. The statistics are shown in Table 1.

5.2.1. Results and discussion

In the first experiment, we compare the different enrollment and diarization methods. From each speaker's eight enrollment calls, we create one enrollment model from the first four calls and one enrollment model from the last four calls so that we have two enrollment models per speaker. The latter is not particularly realistic from a scenario point of view, since it means the user would use the fifth to the eight intercepted call for creating the enrollment model. However, using the data in this way, doubles the number of test trials compared to the amount given in Table 1, i.e., we have 360 target trials and 3288 non-target trials. Since the number of trials are few, we believe this is more important then following the most realistic scenario of the dataset. The results are in Tables 2 and 3. The most noticeable observations are

- The Complete cluster search with PLDA likelihood objective is the best in most cases.
- It is better to apply preprocessing before the mean operation.
- The intersection method is comparable to the Cluster method
- Generally, PCA diarization works better than VBx both for enrollment and verfication.

It is surprising that the PCA diarization method performed better than VBx. However, it should be noted that VBx has many tunable parameters that we did not explore (we used the default settings in the recipe). Also, the methods for constraining the number of speakers may not be the best.

In the second experiment, we analyse the impact of the number of enrollment calls. Based in the conclusions from the previous experiment, we consider only PCA based diarization in both enrollment and verification. For the Cluster method, we use only PLDA score with preprocessing before the mean operation for the enrollment embeddings. The results are in Tables 4 and 5. From these results we can see that the median method can be a good choice if few enrollment calls are available. The results of the median method may seem a bit random, changing between 12.2 and 5.0. This is, however, not strange considering the nature of the median method. Due to channel effects, it may happen that a given embeddings from the same call are located close to each other. The median may jump between such call specific clusters when more enroll data is added.

Table 2: EER (%) for distinct enrollment and verification combinations using PLDA mean scoring. "Cluster_[std—plda]" refers to the cluster method with average standard deviation or PLDA likelihood as objective respectively. "mean-pp" means that embeddings for enrollment were averaged before the preprocessing which is analogous to the Median and Intersection method. "pp-mean" means that the embeddings were preprocessed before averaging which is the standard way in PLDA scoring.

	PCA	VBx
Median	8.89	11.11
Intersection	5.00	7.78
Cluster_std, pca_diar, mean-pp	19.44	18.61
Cluster_plda, pca_diar, mean-pp	3.89	6.11
Cluster_std, VBx_diar, mean-pp	11.11	13.61
Cluster_plda, VBx_diar, mean-pp	5.00	6.94
Cluster_std, pca_diar, pp-mean	18.61	18.61
Cluster_plda, pca_diar, pp-mean	**3.61**	**5.56**
Cluster_std, VBx_diar, pp-mean	11.67	13.06
Cluster_plda, VBx_diar, pp-mean	4.17	6.39

Table 3: . EER (%) for distinct enrollment and verification combinations using PLDA by-the-book scoring. See Table 2 for explanations of the notations.

	PCA	VBx
Median	9.44	11.39
Intersection	5.28	7.78
Cluster_std, pca_diar, mean-pp	18.33	18.06
Cluster_plda, pca_diar, mean-pp	4.17	6.39
Cluster_std, VBx_diar, mean-pp	11.67	13.89
Cluster_plda, VBx_diar, mean-pp	5.28	7.50
Cluster_std, pca_diar, pp-mean	16.67	17.22
Cluster_plda, pca_diar, pp-mean	**3.33**	5.56
Cluster_std, VBx_diar, pp-mean	11.67	12.50
Cluster_plda, VBx_diar, pp-mean	3.89	**5.28**

It should be noted that due to the small number of trials, the difference between the intersection and the cluster method is most likely not significant. However, the main point of this experiment is to provide some understanding of how the methods depend on the number of enrollment calls.

5.3. Data from real case

We further evaluate the proposed methods using conversations obtained from a real case[5]. The database consists of embedding matrices of close to 200 mono conversations with varied lengths (see Fig. 3), labeled with speakers pin code. We use speakers

[5] The recordings were fully anonymized to the GDPR and national law standard and provided as x-vector matrices by a law enforcement agency to the Roxanne consortium. The standard of anonymization was checked by technical experts and legal advisers.

Table 4: EER (%) for distinct enrollment and verification combinations using PLDA mean scoring.

	2	4	6	8
Median	11.11	5.00	3.89	12.78
Intersection	4.44	3.33	2.78	3.33
Cluster	18.89	5.00	2.22	1.67

Table 5: EER (%) for distinct enrollment and verification combinations using multisession PLDA by-the-book scoring.

	2	4	6	8
Median	12.22	5.00	5.00	12.22
Intersection	4.44	3.33	2.78	2.78
Cluster	19.44	5.00	1.67	1.67

with more than two conversations as potential targets, so that we can build a minimal model with two conversations and test in the remaining one. From a total of 87 unique speakers in the database, 25 speakers qualify under this criterion. The other speakers (with less than three conversations) participate both as extra impostors for the target speakers or as a cohort set for t-norm [13] score normalization. Cohort and Impostor set compositions are slightly different for each trial, since the cohort must not contain conversations in which either model or testing speaker participate and, in addition, both testing speakers in an impostor conversation must be different from the model. Moreover, the number of conversations of each of the 25 target speaker is highly imbalanced (see the distribution in Figure 4). Therefore to comply with this real scenario and due to the limited amount of data, the evaluation is performed through one-leave-out cross-validation, i.e., for each target speaker, we use all his available conversations for building models, except the testing one. At all, there are 525 target trials and 6950 impostor trials.

5.3.1. Results

Table 6 shows evaluation results for six enrollment/verification combinations using the methods described earlier. Performance in EER (%) for each of the *Median*, *Intersection* and *Clustering* enrollment methods are presented for either *PCA* and *VBx* verification methods, using cosine similarity scoring optionally followed by t-norm. The results clearly indicate the superiority of the *Intersection* enrollment method and suggests that there's not much difference concerning the type of segmentation for testing conversation. This observation is in line with similar experiments using NIST benchmarks also suggesting that more refined testing segmentation does not necessarily increase speaker recognition performance [7]. Another interesting point is the huge improvement t-norm adds to performance, comparing to the improvement in regular single-channel conversation benchmarks. The explanation seems to be that scores obtained for the segmented speakers in the testing conversation are differently biased and it is important to reduce this effect before the *max* operator (see Figure 2 and Equation 6). We also investigated the effects of PLDA scoring on this evaluation and results are shown in Table 7. We noted that t-norm score normalization decreased performance (and results are omitted), possibly because bias suppression is inherent in PLDA scoring. Moreover, we observe that the *Intersection* and *Clustering* enrollment methods are quite competitive.

Table 6: EER (%) for distinct enrollment and verification combinations using cosine similarity.

	PCA	PCA+tnorm	VBx	VBx+tnorm
Median	21.4	17.3	20.6	17.4
Intersection	14.7	9.0	15.6	9.3
Cluster	18.3	14.7	19.2	15.4

Table 7: EER (%) for distinct enrollment and verification combinations using PLDA.

	PCA	VBx
Median	16.8	18.3
Intersection	14.1	16.8
Cluster	14.2	15.6

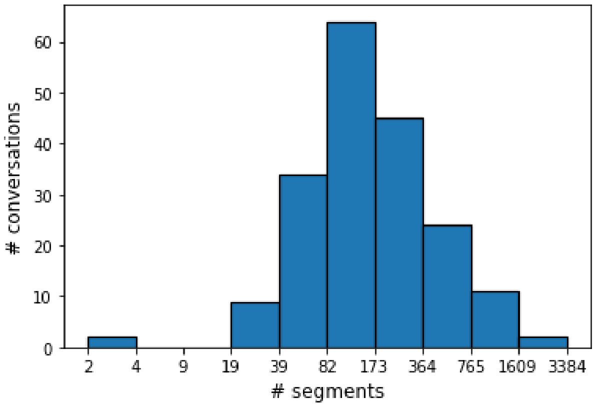

Figure 3: Conversations length in terms of number of segments.

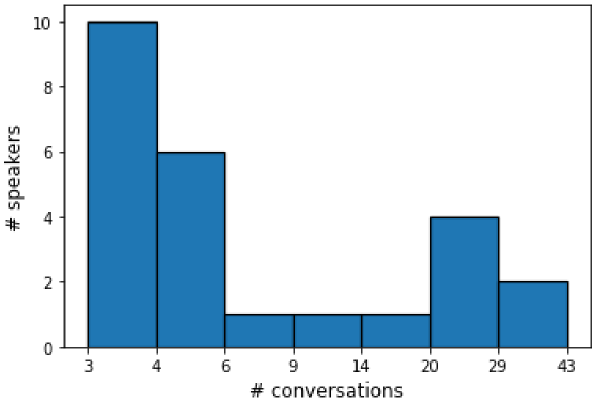

Figure 4: Number of conversations per target speaker.

6. Conclusions and Future work

This paper approached the task of speaker enrolling and verification on multi-speaker (mono) recording scenarios. Specifically, we assume that at least a few conversations of a target speaker are available for building a model which is used to search this speaker in other conversations. Methods for both enrollment and verification are proposed and evaluated using mock and real databases. We show that even simple methods not requiring tunable settings can perform well in these challenging and unpredicted scenarios. Nevertheless, bigger databases should be used to confirm these findings. The methods proposed can be naturally extended to more than two speakers in a single channel. Furthermore, more refined embedding averaging schemes can be used. Those will be the focus of future research.

7. Acknowledgments

The work was supported by European Union's Horizon 2020 project No. 833635 ROXANNE. Computing on IT4I supercomputer was supported by the Czech Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project "e-Infrastructure CZ – LM2018140".

8. References

[1] Mitchell McLaren, Luciana Ferrer, Diego Castán, and Aaron D. Lawson, "The speakers in the wild (sitw) speaker recognition database," in *INTERSPEECH*, 2016.

[2] "Nist 2018 speaker recognition evaluation plan," `https://www.nist.gov/system/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf`, 2018.

[3] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5796–5800.

[4] Martin Karu and Tanel Alumäe, "Weakly supervised training of speaker identification models," in *Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 24–30.

[5] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[6] Hagai Aronowitz, "Unsupervised compensation of intra-session intra-speaker variability for speaker diarization," in *Odyssey 2010 The Speaker and Language Recognition Workshop*, 2010, pp. 138–145.

[7] Yosef A. Solewicz and Hagai Aronowitz, "Implicit segmentation in two-wire speaker recognition," in *Proc. Interspeech 2011*, 2011, pp. 377–380.

[8] Hagai Aronowitz, Yosef Solewicz, and Orith Toledo-Ronen, "Online two speaker diarization," in *Odyssey 2012 The Speaker and Language Recognition Workshop*, 2012, pp. 138–145.

[9] Federico Landini, Ján Profant, Mireia Diez, and Lukáš Burget, "Bayesian hmm clustering of x-vector sequences (vbx) in speaker diarization: theory, implementation and analysis on standard tasks," *Computer Speech & Language*, vol. 71, pp. 101254, 2022.

[10] Sergey Ioffe, "Probabilistic linear discriminant analysis," in *ECCV*, 2006.

[11] Johannes Traa, "Least-squares intersection of lines," *University of Illinois Urbana-Champaign (UIUC)*, 2013.

[12] Kvetoslav Maly, Gerhard Backfried, Francesco Calderoni, Jan "Honza" Černocký, Erinc Dikici, Maël Fabien, Jan Hořínek, Joshua Hughes, Miroslav Janošík, Marek Kovac, Petr Motlicek, Hoang H. Nguyen, Shantipriya Parida, Johan Rohdin, Miroslav Skácel, Sergej Zerr, Dietrich Klakow, Dawei Zhu, and Aravind Krishnan, "ROXSD: a Simulated Dataset of Communication in Organized Crime," in *Proc. 2021 ISCA Symposium on Security and Privacy in Speech Communication*, 2021, pp. 32–36.

[13] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1, pp. 42–54, 2000.