Hierarchical speaker clustering methods for the NIST i-vector Challenge

Elie Khoury, Laurent El Shafey, Marc Ferras, Sébastien Marcel

Idiap Research Institute
Martigny, Switzerland
{ekhoury,lelshafey,mferras,marcel}@idiap.ch

Abstract

The process of manually labeling data is very expensive and sometimes infeasible due to privacy and security issues. This paper investigates the use of two algorithms for clustering unlabeled training i-vectors. This aims at improving speaker recognition performance by using state-of-the-art supervised techniques in the context of the *NIST i-vector Machine Learning Challenge 2014*. The first algorithm is the well-known Ward clustering that aims at optimizing an objective function across all clusters. The second one is a cascade clustering, which benefits from the latest advances in speaker modeling and session compensation techniques, and relies on both the cosine similarity and probabilistic linear discriminant analysis (PLDA). Furthermore, this paper investigates the multi-clustering fusion that opens the door for further improvements. The experimental results show that the use of the automatically labeled i-vectors to train supervised methods such as LDA, PLDA or linear logistic regression-based fusion, decreases the minimum decision cost function by up to 22%.

1. Introduction

Modern speaker recognition systems typically rely on several handcrafted preprocessing or feature extraction techniques, and involve speech corpus engineering. This required knowledge often prevents researchers outside the audio processing community to be involved in speaker recognition evaluations (SREs). To foster the interest of a wider range of researchers and, *e.g.*, the application of recent advances in machine learning, NIST has organized a novel benchmark, the *NIST i-vector Machine Learning Challenge 2014*.[1] In contrast to previous NIST SREs, this challenge relies on the i-vector paradigm [1], which is widely used by state-of-the-art speaker recognition systems [2]. By providing such i-vectors directly instead of audio data, this benchmark is accessible to participants outside the audio processing community.

This i-vector paradigm only acts as a front-end, by extracting low-dimensional i-vectors from speech utterances of varying durations. Many session compensation and classification techniques commonly applied on top of i-vectors are supervised, such as probabilistic linear discriminant analysis (PLDA) [3], linear discriminant analysis (LDA) and within-class covariance normalization (WCCN) [1]. They hence require a labeled and preferably large training set. Similarly, both true claimant and impostor trials, and, hence, labeled training data are required to fuse several speaker recognition systems. In contrast, the development data provided by NIST for this competition are unlabeled. Therefore, one of the main challenges of this benchmark is how to effectively use the unlabeled i-vectors from the development set.

A potential solution to this problem is *clustering*, grouping the unlabeled i-vectors into several clusters, before assigning a label to each cluster. The resulting automatically labeled i-vectors can then be employed as a training set for state-of-the-art supervised classification and fusion techniques. However, the clustering technique must be totally unsupervised, which is not the case in many existing approaches [4, 5]. In this work, we evaluate two fully unsupervised bottom-up clustering approaches. The first one is the Ward clustering [6, 7] that aims at optimizing an objective function across all clusters. The second one is a two-step iterative technique, which employs two different similarity measures. The first step merges clusters using the cosine metric between the average i-vectors representing each cluster. The resulting clusters are used to train a PLDA model, which enables a second clustering step based on PLDA scoring.

The evaluation of these two clustering techniques is performed by studying their impact on two different front-end speaker recognition systems: (1) a LDA-Cosine system based on the baseline provided by NIST that relies on *cosine scoring* (also known as *fast scoring*), and (2) a *PLDA*-based system.

The remainder of the paper is organized as follows. Section 2 presents the related work. Section 3 briefly reviews the two front-end speaker recognition systems used in our experiments. Section 4 presents both clustering algorithms and their fusion using the Hungarian algorithm. The experimental results are shown in section 5. Section 6 concludes the paper.

2. Related Work

2.1. Speaker Recognition

The data supplied by NIST for this challenge are i-vectors that were extracted using a speaker recognition system developed by the John Hopkins University Human Language Technology Center of Excellence in conjunction with MIT Lincoln Laboratory for the 2012 NIST SRE.[2] This i-vector paradigm [1] is built on top of the Gaussian mixture model (GMM) framework [8]. It aims at extracting a low-dimensional factor $\boldsymbol{w}$, called an i-vector, from a speech utterance $\mathcal{O}$. This approach has been successfully and widely used by the speaker recognition community [1, 9, 10, 11, 2]. It relies on the definition of a low-dimensional total variability subspace $\boldsymbol{T}$ and can be described in the GMM mean supervector space by:

$$\boldsymbol{\mu} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{w}\,, \tag{1}$$

where $\boldsymbol{\mu}$ is the GMM mean supervector that best describes the sample $\mathcal{O}$, $\boldsymbol{w}$ is the low-dimensional i-vector extracted from

[1] http://www.nist.gov/itl/iad/mig/ivec.cfm

[2] http://www.nist.gov/itl/iad/mig/sre12.cfm

the sample $\boldsymbol{\mathcal{O}}$, which is assumed to follow a normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and $\boldsymbol{m}$ is the GMM mean supervector of the universal background model (UBM). In practice, $\boldsymbol{T}$ is learned using the expectation-maximization (EM) algorithm.

This i-vector approach only acts as a front-end extractor and does not perform session compensation or scoring. Therefore, several techniques are commonly applied to i-vectors.

The authors in [12] show that whitening and length-normalization are very helpful preprocessing techniques. Whitening consists of normalizing the i-vector space, such that their covariance matrix is turned into the identity matrix. Length-normalization aims at reducing the mismatch between training and testing i-vectors by projecting them into a unit sphere.

Several scoring techniques have been employed to classify i-vectors. A simple and fast approach is cosine scoring [1]. Prior to scoring, linear discriminant analysis (LDA) can be applied to boost the performance. Another popular technique is probabilistic linear discriminant analysis (PLDA), which is a supervised generative framework initially proposed for the task of face recognition [3]. Several variants of this model have been proposed. Heavy-tailed prior distributions can be used instead of Gaussian ones [9, 11]. Besides, fully Bayesian formulations have been proposed in [13, 10]. In this work, we rely on the original approach with Gaussian priors, for which a scalable and exact formulation exists [14, 15].

On the other hand, the fusion of speaker recognition systems often allows significant performance improvements [2]. Empirical approaches such as the sum or majority voting rules can be employed for this purpose. However, better performance is often achieved when using true claimant and impostor scores to train a supervised classifier for the fusion. Such a simple and efficient approach relies on logistic regression [16].

2.2. Speaker Clustering

When employing state-of-the-art supervised scoring or fusion techniques, a large labeled training set is required. In contrast, the development set supplied for this challenge is unlabeled. Clustering is a possible way to circumvent this problem.

Before the development of the i-vector paradigm, speaker clustering using supervectors was explored in [17, 18]. In their work, the authors found that the simple cosine similarity measure is very effective at clustering utterances that belong to the same speaker, when applied to supervectors, which are dimensionality reduced using principal component analysis (PCA). More recently, a similar finding was shown using i-vectors [19, 20, 21, 22]. In [19], a cosine-based k-means clustering is used to group telephone speech conversations in which the number of speakers is *a priori* known ($k = 2$). A more generalized version of this approach is proposed in [20] using spectral clustering along with a simple heuristic that aims at automatically finding the number of speakers. In [21, 22], the authors successfully extended the mean shift approach [23] by employing the cosine similarity instead of the Euclidean distance.

Other approaches that do not rely on the cosine similarity were also investigated. A probabilistic approach is proposed in [24], applying a Bayesian GMM to PCA-reduced i-vectors. In [5], speaker clustering is redefined as an integer linear programming (ILP) problem where the goal is to find a global optimum over the whole clusters. A labeled training set is then required to compute the within-class covariance matrix used in their distance metric. In [4], a PLDA-based clustering shows a good improvement over the well-known Bayesian information criterion (BIC) algorithm [25]. Once more, this approach requires a labeled training set to estimate the PLDA model.

3. Speaker Recognition

In this section, we describe the two different scoring strategies employed in this work. The first one relies on the NIST baseline, which employs cosine scoring, but an LDA step has been added. The second one is the popular PLDA approach.

3.1. LDA-Cosine System

As part of the i-vector challenge, NIST provides an implementation of a baseline system, which is a variant of cosine scoring. The following system is based on this approach, but makes use of an additional LDA step to boost the performance.

First, i-vectors are preprocessed. The sample mean and the sample covariance of the unlabeled development data are computed. These statistics are then used to center and whiten the i-vectors. Next, length-normalization is performed, projecting all the i-vectors into the unit Euclidean sphere.

Moreover, when a labeled training set is available, it has been shown that LDA applied prior to cosine scoring is a very effective strategy [1].

At enrollment time, for each target model i, its $J_i = 5$ length-normalized enrollment i-vectors $\boldsymbol{w}_{i,j}$ are averaged:

$$\boldsymbol{w}_i = \frac{1}{J_i} \sum_{j=1}^{J_i} \boldsymbol{w}_{i,j} \,, \tag{2}$$

a strategy, which was empirically shown to be efficient in [26]. The resulting client models $\boldsymbol{w}_i$ are then projected into the unit sphere, leading to $\boldsymbol{w}_i$.

At test time, the cosine similarity measure [1] is employed. This is a simple and efficient method for estimating how close an i-vector $\boldsymbol{w}_t$ extracted from a test utterance $\boldsymbol{\mathcal{O}}_t$ is to the average i-vector $\boldsymbol{w}_i$ representing a target speaker i:

$$h_{\text{cosine}}(\boldsymbol{w}_i, \boldsymbol{w}_t) = \frac{\boldsymbol{w}_i \cdot \boldsymbol{w}_t}{\|\boldsymbol{w}_i\| \|\boldsymbol{w}_t\|} = \boldsymbol{w}_i \cdot \boldsymbol{w}_t \,, \tag{3}$$

the i-vectors being length-normalized ($\|\boldsymbol{w}_i\| = \|\boldsymbol{w}_t\| = 1$).

3.2. PLDA System

Recently, another technique called PLDA [3] has been shown to be very efficient when applied to i-vectors [2]. PLDA is a supervised, generative and probabilistic framework that models both speaker and channel effects.

More formally, PLDA assumes that the j^{th} i-vector of speaker i is generated by:

$$\boldsymbol{w}_{i,j} = \boldsymbol{F}\boldsymbol{h}_i + \boldsymbol{G}\boldsymbol{k}_{i,j} + \boldsymbol{\epsilon}_{i,j} \,, \tag{4}$$

where $\boldsymbol{F}$ and $\boldsymbol{G}$ are the subspaces describing the speaker and channel effects, respectively. $\boldsymbol{h}_i$ and $\boldsymbol{k}_{i,j}$ are the associated latent variables, which are assumed to be normally distributed $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Finally, $\boldsymbol{\epsilon}_{i,j}$ represents the residual noise, which is supposed to follow a Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon)$.

The parameters $\boldsymbol{\Theta} = \{\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{\Sigma}_\epsilon\}$ of this model are learned using an expectation-maximization algorithm over a labeled training set of i-vectors. In this work, this labeled training set is obtained using clustering techniques (*cf.* section 4). Once the model has been trained, given a test i-vector $\boldsymbol{w}_t$ and an i-vector

$\boldsymbol{w}_i$ representing a target speaker i, authentication is achieved by computing the following log-likelihood ratio (LLR) score:

$$h_{\text{plda}}\left(\boldsymbol{w}_i, \boldsymbol{w}_t\right) = \ln\left(\frac{p(\boldsymbol{w}_i, \boldsymbol{w}_t \mid \boldsymbol{\Theta})}{p(\boldsymbol{w}_i \mid \boldsymbol{\Theta})p(\boldsymbol{w}_t \mid \boldsymbol{\Theta})}\right). \quad (5)$$

Here, $p(\boldsymbol{w}_i, \boldsymbol{w}_t \mid \boldsymbol{\Theta})$ is the likelihood that the i-vectors $\boldsymbol{w}_i$ and $\boldsymbol{w}_t$ share the same latent identity variable $\boldsymbol{h}_i$ and, hence, are coming from the same target speaker. In contrast $p(\boldsymbol{w}_i \mid \boldsymbol{\Theta})p(\boldsymbol{w}_t \mid \boldsymbol{\Theta})$ is the likelihood that the i-vectors $\boldsymbol{w}_i$ and $\boldsymbol{w}_t$ have different latent identity variables $\boldsymbol{h}_i$ and $\boldsymbol{h}_t$ and, therefore, are from different speakers.

We employ the same preprocessing and enrollment strategies as for the cosine scoring baseline (*cf.* section 3.1). In particular, this means that a target speaker model is generated by averaging its enrollment samples (*cf.* Eq. (2)).

For details on how to train the parameters $\boldsymbol{\Theta}$ and how to estimate the likelihood, readers are referred to [3, 14].

4. Hierarchical Clustering

4.1. Ward Clustering

We use the so-called Ward clustering [6] as the first choice to map i-vectors to speakers in an unsupervised manner. This is a greedy agglomerative clustering algorithm that, in its general form, aims at optimizing an overall objective function. In this work, the sum of within-class scatter of the optimal partition, *i.e.*, across all clusters, is minimized, resulting in maximally compact groups of i-vectors. For this objective function, the so-called Lance-Williams [27] algorithm computes distances between pairs of clusters recursively, reducing the clustering time drastically. Both Ward and Lance-Williams algorithms fit together when square Euclidean derived distances are used, resulting in the algorithm described below.

Initially, each i-vector is assigned a different cluster as is typically done in bottom-up hierarchical approaches to clustering. The Euclidean distance between all pairs of i-vectors is computed and stored in a distance matrix.[3] The two closest clusters are successively merged until only one remains, obtaining the whole clustering dendrogram as output.

A key point of agglomerative clustering algorithms is the linking method, or how to measure the distance between two clusters at some stage in the clustering process. Instead of directly recomputing the distance between two clusters of i-vectors at each iteration, the Lance-Williams [27] recursion enables the computation of all the distances required at any clustering step from the initial distance matrix. Assuming two clusters C_i and C_j are being merged, the distances between the merged cluster $C_{(ij)}$ and all other clusters C_k are updated as:

$$d_{(ij)k} = \alpha_i d_{ik} + \alpha_j d_{jk} + \beta d_{ij}\,, \quad (6)$$

with d_{ij} being the distance from cluster C_i to cluster C_j and:

$$\alpha_i = \frac{n_i + n_k}{n_i + n_j + n_k}\,, \quad (7)$$

$$\alpha_j = \frac{n_j + n_k}{n_i + n_j + n_k}\,, \quad (8)$$

$$\beta = \frac{n_k}{n_i + n_j + n_k}\,, \quad (9)$$

and n_l being the number of i-vectors in cluster C_l. In this work, each i-vector is initially in a separate cluster. Other choices for α_i, α_j and β lead to *single, complete* and *average linkage*.

This clustering algorithm is a simplified version of the one presented in [7], with the difference that the Euclidean distance is used instead of the Hotelling t-square distance. The factor covariance matrices used in the computation of the total factors are not available in the data provided for the NIST i-vector challenge.

Speaker clusters are expected to naturally arise during the clustering process. We assume the speaker clusters can be simply found by thresholding the distance values in the clustering dendrogram. For a given parent node p and child node c in the dendrogram, if $d_p > \theta_1$ and $d_c < \theta_1$, all descendants including node c are assigned the same speaker identifier.

4.2. Cosine-PLDA Clustering

The second choice to map i-vectors to speakers in an unsupervised bottom-up manner relies on the use of two similarity measures that are commonly employed in the speaker recognition field when dealing with i-vectors: (1) the cosine similarity measure [1] and (2) the PLDA similarity measure [9, 11]. The former (*i.e.*, cosine measure) does not require any training data, which makes it very suitable to our clustering problem. The latter (*i.e.*, PLDA measure) has been shown to be better but needs training data to learn the PLDA model. To take advantage of both measures, we propose a cascade system with two main clustering steps: (1) cosine-based clustering and (2) PLDA-based clustering. This clustering is illustrated in Fig. 1.

Step 1: Cosine-based clustering. As for Ward clustering, a symmetric similarity matrix is computed using cosine measure between each pair of i-vectors. As quoted in [22], "a rationale for using cosine similarity instead of Euclidean distance can be supplied by postulating a normal distribution for the speaker population". Once the similarity matrix is computed, the closest clusters are merged iteratively until a stopping criterion is reached. In practice, the stopping criterion θ_2 is set reasonably high[4] to ensure pure clusters for the next clustering step. The update of the similarity matrix at each iteration is performed by computing cosine measure between the average i-vectors of the updated clusters (*cf.* Eq. 3). The choice of this special kind of linkage is motivated by both the study in [26] and our preliminary experiments on this i-vector challenge. Both lead to the same conclusion that averaging the length-normalized i-vectors is always better than computing the average, maximum and minimum scores. This explains why the *average*, *complete* and *single linkages* -which are less expensive than our proposed linkage- are discarded in this algorithm.

Step 2: PLDA-based clustering. The first clustering step provides an initial set of clusters that can be used to train an initial PLDA model. In order to reduce the effect of under-clustering (*i.e.*, one speaker belongs to several clusters) of the training set, only clusters with more than four i-vectors are retained. Once the $\boldsymbol{F}$ and $\boldsymbol{G}$ matrices of the PLDA model are estimated, the second step of this clustering is enabled. As for the previous algorithm, a similarity matrix is computed, but this time using the likelihood ratio between each pair of average i-vectors (one average i-vector corresponds to one cluster)

[3] Note that only $N(N-1)/2$ different entries need to be computed, as Euclidean distance is symmetric.

[4] θ_2 was set empirically to 0.4 which corresponds to an angle of around 66° between both average i-vectors.

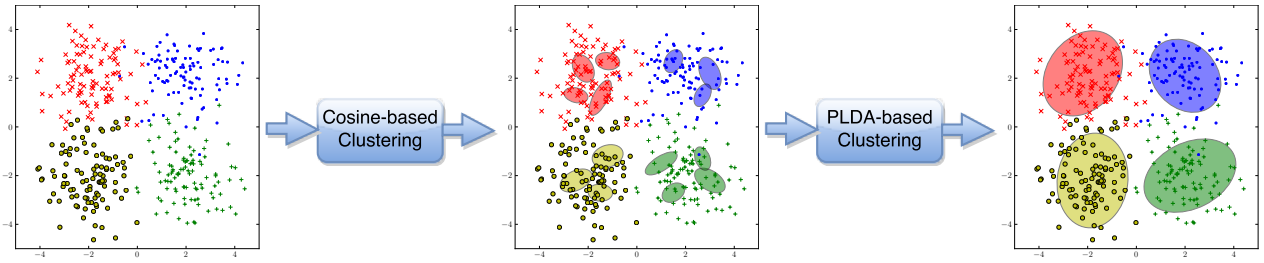

Figure 1: COSINE-PLDA CLUSTERING. *This figure illustrates how this two-step clustering works. At the end of cosine clustering, small clusters are formed. These clusters are used to estimate a PLDA model and thus enable a second clustering based on PLDA.*

as expressed in Eq. (5). Again, it is worth noting that Eq. (5) is symmetric. Therefore, only the upper triangular part of the similarity matrix is computed as for Euclidean and cosine measures. Once the similarity matrix is computed, the closest clusters are merged iteratively until the stopping criterion is reached. Recalling that the log likelihood ratio in Eq. (5) selects one of two hypotheses H_0 (both average i-vectors belong to the same speaker) or H_1 (average i-vectors belong to different speakers), the stopping criterion θ_3 of the hierarchical clustering is theoretically equal to 0.[5]

Regarding the update after each merge of pair of clusters, and since the labeled data of the training set has changed, both the PLDA model and the whole triangular matrix should ideally be re-estimated in contrast to the previous two clustering algorithms (*i.e.*, Ward and cosine clustering) where only one raw (or column) in the matrix needs to be recomputed. In practice, these updates are very costly despite the use of a scalable PLDA implementation. This is why they are re-estimated only every N merges ($N = 500$), and in between two updates, *complete linkage* strategy is applied to avoid any undesired merge.

4.3. Multi-Clustering Fusion

Although the two previously described algorithms share the same bottom-up approach to cluster i-vectors, they rely on different similarity measures and linking strategies. Therefore, they may be complementary and their combination might lead to a performance gain. To study the complementarity of both clustering methods, we use the clustering similarity measure defined in [28]. Let $\boldsymbol{\mathcal{C}} = \{C_1, C_2, \ldots, C_m\}$ and $\boldsymbol{\mathcal{D}} = \{D_1, D_2, \ldots, D_n\}$ denote the sets of clusters provided by both algorithms. The similarity S between $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$ is defined by:

$$S(\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{D}}) = \frac{1}{\max(m,n)} \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{|C_i \cap D_j|}{|C_i \cup D_j|} \,. \qquad (10)$$

A value of S close to 1 indicates that $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$ are very similar, and, thus, there is no potential gain from combining them. If S is close to 0, then, there is no overlap between $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$, which means that at least one of them is weak. Again, a performance gain cannot be achieved by combining them in this case. The multi-clustering fusion could be helpful only when S is far from these extremes.

In our case, we found that $S = 0.54$ when $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$ are selected at the operating point that minimizes the *minimum decision cost function* (minDCF) on the progress (*cf.* Fig. 3). This finding keeps the door open to further improvements, and motivates us to investigate several combination methods. In this work, we explore the Hungarian algorithm [29], a combinatorial optimization algorithm that aims at finding the best mapping between both sets of clusters.

The Hungarian algorithm was used in [30] to associate speaker and face clusters for bimodal person diarization. Similarly, we use this strategy to associate the two sets of speaker clusters. First, the co-occurrence matrix between $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$ is computed. Next, pairs of clusters with the highest overlap are iteratively selected. At each iteration, columns and rows corresponding to these pairs are filled with zeros. The process ends when the whole co-occurrence matrix turns to zero.

This algorithm results in a new set of clusters $\boldsymbol{\mathcal{F}} = \{F_1, F_2, \ldots, F_p\}$, with $p \leq \min(m, n)$, that have always same or higher purity (*i.e.*, less outliers) than the ones in $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$. Therefore, it forms a confident training set for supervised recognition systems. However, when both clustering are not at the same level of performance, there is a risk of "chunking" a good cluster, which would produce a less optimal training set for supervised techniques. To reduce such a risk, $\boldsymbol{\mathcal{F}}$ is only used to train the linear logistic regression-based score fusion of the two recognition systems, LDA-Cosine and PLDA. In section 5, this strategy is compared with the ones that solely rely on $\boldsymbol{\mathcal{C}}$ or $\boldsymbol{\mathcal{D}}$ as training sets for score fusion.

A summary of the strategies employed in this work is depicted in Fig. 2.

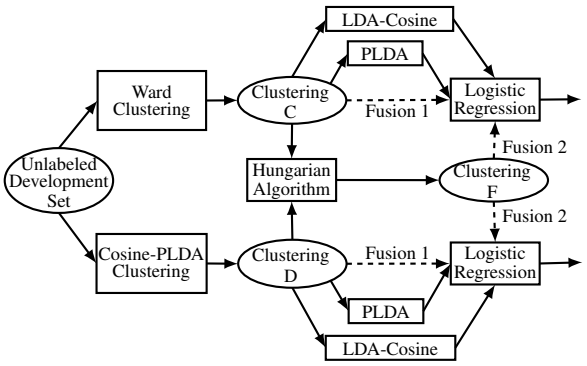

Figure 2: SUMMARY OF THE SPEAKER RECOGNITION SYSTEMS. *This figure illustrates how the different clustering techniques are applied to build several speaker recognition systems.*

[5] In practice, better results in terms of minDCF on the progress set are achieved with $\theta_3 \approx 20$ using our PLDA implementation [14].

5. Experimental Results

Two main sets are provided by NIST: development and evaluation sets. The development set comprises $36,572$ unlabeled i-vectors. The evaluation set includes both enrollment and test data. The enrollment data consist of $1,306$ target speaker models (both female and male) with 5 i-vectors per models. The test data comprise $9,634$ i-vectors. There are $12,582,004$ trials that consist of all possible pairs involving a target model and a test i-vector. The trials are divided into two subsets: progress and evaluation subsets, comprising 40% and 60%, respectively. The results on the progress set are provided to participants just after their submission. The performance metric used in this evaluation is the minDCF, which is defined by:

$$\text{minDCF} = \min_t \left(\text{FRR}(t) + 100\,\text{FAR}(t) \right), \qquad (11)$$

where FRR and FAR denote the false rejection rate and false acceptance rate, and t the varying threshold.

The experiments in this work are conducted using SPEAR [31],[6] a new speaker recognition python library developed upon Bob [32]. The PLDA parameters D_F, D_G and the number of iterations are set to 300, 100 and 200, respectively. The dimension of the LDA projection matrix is set to 300.

5.1. Impact of Clustering Algorithms

First we evaluate the impact of both Ward and Cosine-PLDA clusterings on the recognition performance, throughout the agglomerative clustering process. This is performed using the PLDA based recognition system (section 3.2). Fig. 3 draws the minDCF when decreasing the number of clusters from $30k$ to $10k$. This figure shows that both clusterings are able to outperform the NIST baseline system, and that Cosine-PLDA clustering achieves a relative decrease of about 22% in minDCF when compared to the baseline system. This minDCF value is obtained for a number of clusters equal to $16k$. This leads in practice to a training set of only 1942 clusters because only clusters with more than four i-vectors are feeding the PLDA. Fig. 3 also shows that the Cosine-PLDA clustering is more adequate to the problem of speaker recognition than the Ward clustering when dealing with i-vectors. However, it is worth noting that Ward clustering is much faster than Cosine-PLDA clustering, since the latter requires costly updates of the PLDA model and the similarity matrix as described in section 4.2.

Similar trends are observed on the experiments conducted on the LDA-Cosine system. Table 1 summarizes these findings. This table shows that the LDA-Cosine system outperforms the baseline by around 8% when using the Cosine-PLDA clustering, but it is worse than the PLDA recognition system.

5.2. Impact of Multi-Clustering Fusion

The last two columns of Table 1 shows the results of combining both Cosine+LDA and PLDA systems, and this using two different strategies: Fusion 1 and Fusion 2. As illustrated in Fig. 2, Fusion 1 uses the same training set that was initially used to learn the LDA/PLDA parameters, whereas Fusion 2 uses the training set provided by the Hungarian algorithm. Table 1 shows that the results of Fusion 1 are always worse than the best unimodal system (*i.e.*, PLDA system). However, Fusion 2 provides an improvement for the systems based on Ward clustering by reducing the minDCF from 0.355 to 0.345, which leads to relative decrease of around 3% in minDCF. Regarding the systems based on Cosine-PLDA clustering we have noticed neutral effect of Fusion 2. This is mainly due to the fact LDA-Cosine and PLDA systems are not at the same level of performance for this clustering.

[6] https://pypi.python.org/pypi/bob.spear

Table 1: PERFORMANCE SUMMARY. *This table reports the minDCF of the different systems on the progress set.*[7]

Clustering	LDA + Cosine	PLDA	Fusion 1	Fusion 2
Ward	0.374	0.355	0.357	**0.345**
Cosine-PLDA	0.356	**0.300**	0.302	**0.300**

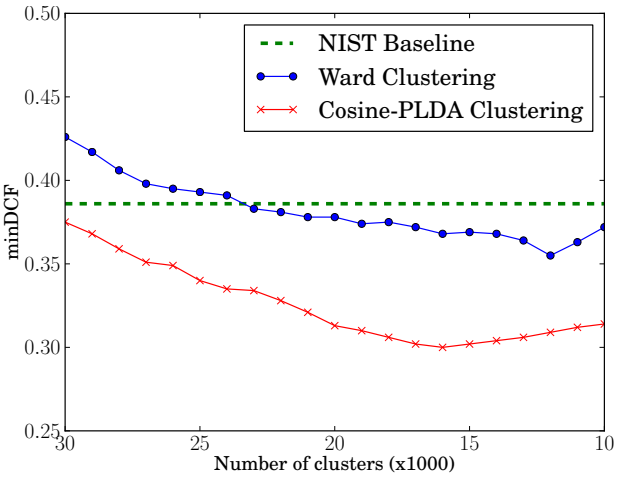

Figure 3: MINDCF IN TERMS OF THE NUMBER OF CLUSTERS. *This figure shows the minDCF values computed by NIST on the progress set in terms of the number of the clusters for both clustering methods on the PLDA recognition system.*

6. Conclusions

In this paper, we investigated the use of two algorithms for clustering unlabeled training i-vectors to improve speaker recognition performance in the context of the *NIST i-vector Machine Learning Challenge 2014*. These clustering techniques allow the use of state-of-the-art supervised techniques such as LDA, PLDA or linear logistic regression-based fusion. The first one is the Ward clustering, which is a very fast approach. The second one is a cascade clustering, which relies on both the cosine similarity and PLDA, and decreases the minimum decision cost function by up to 22%. We also explored the use of Hungarian algorithm for multi-clustering fusion for which promising results are obtained. Future work might consider semi-supervised approaches to benefit from both hand-labeled and large unlabeled pools of data.

7. Acknowledgment

The research leading to these results has received funding from the Swiss National Science Foundation (SNSF) under the LOBI project and from the European Community's Seventh Framework Programme (FP7) under grant agreement 284989 (BEAT) and grant agreement 287872 (inEvent).

8. References

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

[2] R. Saedi, K. A. Lee, T. Kinnunen, et al., "I4U submission to NIST SRE 2012: a large-scale collaborative effort for noise-robust speaker verification," in *INTERSPEECH*, 2013.

[3] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE Intl. Conf. on Computer Vision (ICCV)*, 2007, pp. 1–8.

[4] J. Prazak and J. Silovsky, "Speaker diarization using PLDA-based speaker clustering," in *IEEE 6th Intl. Conf. on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, 2011, vol. 1, pp. 347–350.

[5] M. Rouvier and S. Meignier, "A global optimization framework for speaker diarization," in *Odyssey: The Speaker and Language Recognition Workshop*, 2012.

[6] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[7] M. Ferràs and H. Bourlard, "Speaker diarization and linking of large corpora," in *IEEE Spoken Language Technology Workshop (SLT)*, Dec 2012, pp. 280–285.

[8] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[9] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey: The Speaker and Language Recognition Workshop*, 2010.

[10] J. A. Villalba and N. Brümmer, "Towards fully Bayesian speaker recognition: Integrating out the between-speaker covariance," in *INTERSPEECH*, 2011, pp. 505–508.

[11] P. Matejka, O. Glembek, F. Castaldo, M.J. Alam, O. Plchot, P. Kenny, L. Burget, and J. Cernocky, "Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

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

[13] N. Brümmer, "Bayesian PLDA," Tech. Rep., Agnitio Labs, 2010.

[14] L. El Shafey, C. McCool, R. Wallace, and S. Marcel, "A scalable formulation of probabilistic linear discriminant analysis," *IEEE Trans. in Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 7, pp. 1788–1794, 2013.

[15] Y. Jiang, K.-A. Lee, Z. Tang, B. Ma, A. Larcher, and H. Li, "PLDA modeling in i-vector and supervector space for speaker verification.," in *INTERSPEECH*, 2012.

[16] S. Pigeon, P. Druyts, and P. Verlinde, "Applying logistic regression to the fusion of the NIST'99 1-speaker submissions," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 237–248, 2000.

[17] W.-H. Tsai, S.-S. Cheng, Y.-H. Chao, and Wang H.-M., "Clustering speech utterances by speaker using eigenvoice-motivated vector space models," in *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005, vol. 1, pp. 725–728.

[18] E. Khoury, C. Sénac, and R. Andre-Obrecht, "Speaker diarization: Towards a more robust and portable system," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2007, vol. 4, pp. IV–489–IV–492.

[19] S. Shum, N. Dehak, E. Chuangsuwanich, D. A. Reynolds, and J. R. Glass, "Exploiting intra-conversation variability for speaker diarization," in *INTERSPEECH*, 2011.

[20] S. Shum, N. Dehak, and J. Glass, "On the use of spectral and iterative methods for speaker diarization," in *INTERSPEECH*, 2012.

[21] M. Senoussaoui, P. Kenny, P. Dumouchel, and T. Stafylakis, "Efficient iterative mean shift based cosine dissimilarity for multi-recording speaker clustering," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7712–7715.

[22] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE Trans. on Audio, Speech, and Language Processing (TASLP)*, vol. 22, no. 1, pp. 217–227, 2014.

[23] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.

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

[25] S. Chen and P. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the Bayesian Information Criterion," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

[26] P. Rajan, T. Kinnunen, and V. Hautamaki, "Effect of multicondition training on i-vector PLDA configurations for speaker recognition," in *INTERSPEECH*, 2012.

[27] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies. 1. hierarchical systems," *Computer Journal*, vol. 9, pp. 373–380, 1967.

[28] G. J. Torres, R. B. Basnet, A. H. Sung, S. Mukkamala, and B. Ribeiro, "A similarity measure for clustering and its applications," *Intl. Journ. of Electrical, Computer, and Systems Engineering*, vol. 3, no. 3, 2009.

[29] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, 1955.

[30] E. Khoury, C. Sénac, and P. Joly, "Audiovisual diarization of people in video content," *Multimedia Tools and Applications*, vol. 68, no. 3, pp. 747–775, 2012.

[31] E. Khoury, L. El Shafey, and S. Marcel, "Spear: An open source toolbox for speaker recognition based on Bob," in *IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[32] A. Anjos, L. El Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel, "Bob: a free signal processing and machine learning toolbox for researchers," in *20th ACM Intl. Conf. on Multimedia (ACMMM)*, 2012.