aDCF Loss Function for Deep Metric Learning in End-to-End Text-Dependent Speaker Verification Systems

Victoria Mingote, Antonio Miguel, Dayana Ribas, Alfonso Ortega, and Eduardo Lleida

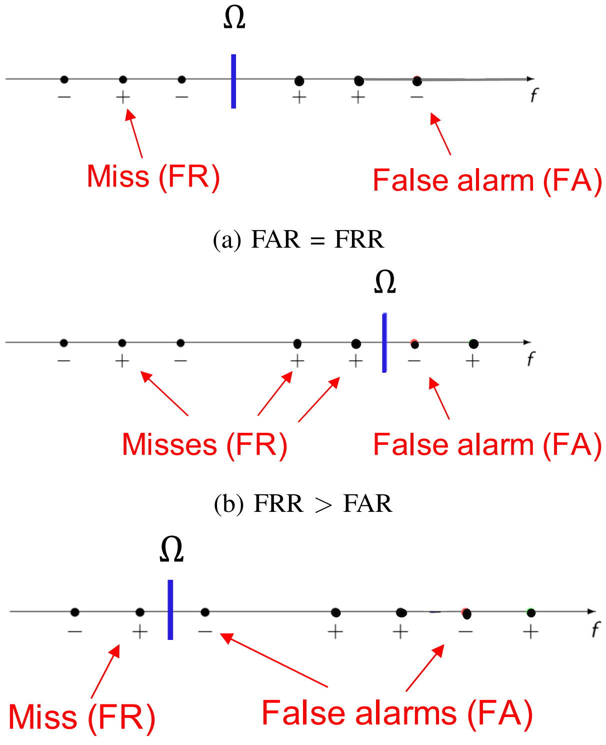
***Abstract*—Metric learning approaches have widely expanded to the training of Speaker Verification (SV) systems based on Deep Neural Networks (DNNs), by using a loss function more consistent with the evaluation process than the traditional identification losses. However, these methods do not consider the performance measure and can involve high computational cost, for example, the need for a careful pair or triplet data selection. This paper proposes the approximated Detection Cost Function (aDCF) loss, which is a loss function based on the measure of the decision errors in SV systems, namely the False Rejection Rate (FRR) and the False Acceptance Rate (FAR). With aDCF loss as the training objective function, the end-to-end system learns how to minimize decision errors. Furthermore, we replace the typical linear layer as the last layer of DNN by a cosine distance layer, which reduces the difference between the metric in the training process and the metric during evaluation. aDCF loss function was evaluated in RSR2015-Part I and RSR2015-Part II datasets for text-dependent speaker verification. The system trained with aDCF loss outperforms all the state-of-the-art functions employed in this paper in both parts of the database.**

***Index Terms*—Speaker Verification, Loss Functions, Metric Learning, aDCF, Cross-Entropy, aAUC, Triplet loss**

I. Introduction

WITH the arrival of the modern artificial intelligence approaches, the automatic recognition of speakers has emerged as one of the most demanded tasks by technological applications. Typically, speaker recognition can be categorized into two modes of operation: speaker identification (SI), and speaker verification (SV). The former consists of assigning a speech sample to a specific speaker identity, in which all the identities are predefined in the training set, while the goal of the latter is to determine whether two speech samples belong to a claimed identity or not. Furthermore, SV systems can be divided into text-independent and text-dependent. In text-independent SV systems, there are no restrictions on the lexicon content, while text-dependent SV systems require the same constraints on the uttered phrase.

The development of SV systems has been an important task during the last decade since the growing interest of related applications such as virtual assistants, home automation, voice authentication systems, among others. SV systems are trained to take a binary decision based on a decision threshold (Ω): acceptance or rejection [1]. Thus, this kind of systems produces two types of decision errors which are depicted in Fig.1. The False Acceptance (FA) is referred to when an impostor speaker is incorrectly accepted (Type I error), and the False Rejection (FR) is related to the incorrect rejection of a true speaker (Type II error) [2] [3]. The system performance is obtained by combining the ratio of these two errors, which are defined by the number of times of each one occurs in relation to the number of legitimate or impostor speakers [4]. The selection of the threshold, Ω, is what relates the system to the operating point of interest in terms of the application, and there are three kind of different cases. Firstly, when FAR $=$ FRR as in Fig.1(a), we are in presence of the Equal Error Rate (EER). This is an operating point frequently used as a measure of the discrimination capability of the system for many applications [5]. However, the EER may not be the best option for some applications, so there are alternative operating



(a) FAR = FRR

(b) FRR $>$ FAR

(c) FRR $<$ FAR

Fig. 1. Decision errors based on the decision threshold (Ω) for SV systems. Different possible cases depending on the amount of each type of error. (a) Case where FAR is equal to FRR. (b) The number of FR are greater than FA. (c) Greater number of FA than FR.

The authors are with ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain (e-mail: {vmingote,amiguel,dribas,ortega,lleida}@unizar.es.)

points [6]. For example, if Ω is set to a high value, the number of false alarms can be decreased even though a greater number of false rejections may occur [7]. Thus, for systems where the cost of accepting an impostor is high, the threshold is chosen such that FRR $>$ FAR as in Fig.1(b). On the other hand, if the application of the system requires a lower number of false rejections, Ω will be set lower, but this involves that the number of impostor speakers accepted increases, so in this case, FRR $<$ FAR as in Fig.1(c). Therefore, due to the relevance of the decision errors in the verification process, we propose the approximated Detection Cost Function (*aDCF*) [8] loss function. This function is based on the measure of these decision errors of SV systems and allows the systems based on deep neural networks (DNN) to be trained directly to optimize a metric focused on the goal task.

Many state-of-the-art SV systems based on DNN [9] [10] follow a typical framework to project the utterances into a low dimensional space. This framework has three key components: a feature acoustic extraction followed by a network front-end, an average pooling mechanism which produces embeddings to represent the whole utterance, and a loss function to train the full system [11] [12]. However, the SV system itself is usually trained using a loss function without encouraging the discriminative learning of features and neither considering the actual operating point. Therefore, when the system provides the performance measure of the whole SV process, this training procedure provokes the system under-performance. Recently, metric learning functions at training have been alternatively used to handle this issue [13] [14]. The main purpose of metric learning algorithms consists of bringing similar samples closer, while different samples are pushed apart from each other using a specific loss function. Thus, these approaches aim to learn a more discriminative embedding space. However, this kind of metric learning loss functions requires careful sample preparation, which usually involves a high computational cost. To address this problem, recent research efforts have been focused on redesigning the traditional classification loss functions to improve the discrimination ability with the same motivation that we developed *aDCF* loss function [8]. Unlike previous approaches, *aDCF* loss function keeps the low computational cost of the traditional classification loss functions, while the system is trained with a metric focused on the goal task to increase the discrimination ability.

In this paper, we study the effects of using *aDCF* loss function combined with a cosine layer as last layer in the DNN architecture instead of a linear layer to train the system. This metric was inspired by the Detection Cost Function (DCF) [15] [16] used by the National Institute of Standards and Technology (NIST) during the Speaker Recognition Evaluations (SRE). This aDCF loss function was proposed to replace the classical Cross-Entropy (CE) loss function [11] [17] [18]. Unlike recent classification or metric learning loss functions [19] [13] [20], aDCF has the ability to adapt the parameters to modify the optimal threshold and the balance between type I and type II errors to meet the requirements of a concrete system application. The ability to manage the scores distributions in function of the operating point is a useful skill to provide for this type of end-to-end systems with. Finally, we show how *aDCF* loss function outperforms the reference systems based on alternative loss functions. Although the proposed method can be used for text-dependent and text-independent SV systems, for the sake of clarity, we present here experimental results on a text-dependent verification task using the well-know RSR2015 dataset. In [8], we presented a preliminary study of *aDCF* loss function, but we did not perform an analysis of the different parameters, and the experimental results shown were limited. For this reason, in this work, we have addressed an extensive analysis of this function and the main contributions are summarized as follows:

- We explore the effects of training SV systems giving different relevance to both types of errors depending on the application.
- We analyze the behaviour of using a complementary loss function in combination with CE loss or *aDCF* loss to improve the discriminative power.
- We compare the performance of *aDCF* loss function against some of the state-of-the-art loss functions.

This paper is laid out as follows. Section 2 presents a review of existing loss functions. The aDCF loss function is presented in Section 3. Section 4 shows the SV system description, followed by the experimental setup in Section 5. Finally, Section 6 discusses obtained results and Section 7 concludes the paper.

II. Loss Functions

Training loss functions play an important role in SV systems based on DNNs, since an effective loss function can improve the discriminative power of the learned features. The selection of which loss function should be used depending on the task is also relevant. Loss functions can be grouped into two categories:

- Identification or classification loss functions: which are used in classification tasks where all the testing identities are predefined in the training set, and features are expected to be separable. In this category, the current extended function is CE loss with softmax output units [11] [17] [18] combined with Ring loss [21] or Center loss [22], and its variants such as angular softmax loss (A-Softmax) [19] or additive angular margin loss (ArcFace) [23].
- Verification or metric learning loss functions: which are designed to improve the discrimination power with a pairwise- or triplet-based training and a similarity metric, which leads a supervised embedding learning in triplet neural network [13] [14], contrastive loss [24], partial AUC loss (pAUC) [25] or NeuralPLDA [26]. Previously [27] [28], we proposed an alternative back-end which combines the triplet loss philosophy with the optimization of the AUC as the loss function (*aAUC*) [29] [30] [31].

Both groups of loss functions have several variants. In this section, we focus on a detailed explanation of the most extended function from each group. We also describe our previous proposed function *aAUC* [27].

A. Cross-Entropy Loss

The traditional and most common identification loss function is CE loss [17] [11] [18]. Due to its simplicity, excellent performance and probabilistic interpretation, this function has been widely applied for multi-class classification. CE loss can be written as,

$$L_{CE} = -\sum_{i}^{m} y_i \cdot \log(\hat{y_i}), \tag{1}$$

where y_i is the ground truth class label with $i \in \{1, ..., m\}$ and m is the number of samples, and $\hat{y}_i$ is the predicted probability extracted from the output of the last fully connected layer.

When a DNN is trained using this loss function, it learns how to separate the features as far away as possible from the decision boundary, which is the goal for a classification task. Nevertheless, deeply learned embeddings are not explicitly encouraged to enlarge the inter-class distance and reduce the intra-class variations, which is not suitable for SV systems. Since they require separable and also discriminative embeddings for the speaker verification task.

B. CE Loss combined with Ring Loss

To solve the previous drawbacks with CE loss and simultaneously keep the same efficiency during training, other approaches have been proposed with encouraging results. CE loss learns to separate embeddings of different classes, but this loss function does not address the intra-class compactness. In order to mitigate the effects of the lack of feature discrimination power, Ring loss [21] was proposed to apply a convex norm constraint over the primary loss to normalize the features and bring compactness to them. Using this complementary function, the system is trained to learn the feature norm close to the unit circle. This contributes to reduce the intra-class variability, while the features can increase their discrimination by using different angles to represent data. Ring loss is formulated as,

$$L_R = \frac{\lambda}{2m} \sum_{i=1}^{m} (||x_i||_2 - R), \tag{2}$$

where R is the target norm value, usually 1, λ is the loss weight, x_i is the input sample of the penultimate layer with $i \in \{1, ..., m\}$ and m is the number of samples. Thus, the joint loss to minimize with this approach is defined as,

$$L = L_{CE} + L_R. \tag{3}$$

C. Angular Softmax Loss

Another alternative has become a reference approach to substitute the traditional softmax loss by redesigning softmax function introducing an angular margin to encourage a larger variance among classes. This loss function is called Angular Softmax or A-Softmax loss [32], which enables the neural networks to learn angular discriminative features. The loss function is defined as follows:

$$\mathrm{L}_{ANG} = -\tfrac{1}{m}\textstyle\sum_{i}^{m} log \frac{\exp\left(||x_i||\psi(\theta_{y_i,i})\right)}{\exp\left(||x_i||\psi(\theta_{y_i,i})\right) + \sum\limits_{j \neq y_i}^{N} \exp\left(||x_i|| cos(\theta_{y_i,j})\right)}, \tag{4}$$

where $\psi(\theta_{y_i,i})$ is the angle function which is a monotonic function and is defined $\psi(\theta_{y_i,i}) = (-1)^k cos(m\theta_{y_i,i}) - 2k$, with $\theta_{y_i,i} \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$, $k \in [0, m-1]$, and m is an integer to control the angular margin.

A-Softmax loss has been proved as an effective method to improve some recognition systems. However, it is difficult to train with this function since it is sensitive to the values of the parameters.

D. Triplet Loss

Motivated by improving the discriminative power of the features extracted from the network, the metric learning approaches or verification loss functions were introduced in the training process of DNNs. One of the most widely verification loss function for metric learning is Triplet loss function [13]. To use this kind of loss function, a triplet neural network structure is applied where three examples are selected to create a negative pair and a positive pair of samples. They are defined as an example from a specific identity called anchor (e), a positive sample of the same identity of the anchor (e^+), and a negative sample from a different identity (e^-). Once the triplet selection process is made, the neural network is trained to enforce a larger similarity metric in the anchor-positive pair than in the anchor-negative pair as Fig.2 depicts. The Triplet loss is defined as,

$$L_{TR} = \sum_{i=1}^{m^+} \sum_{j=1}^{m^-} ||(s_\theta(p_i^+)||_2 - ||s_\theta(p_j^-)||_2 + \tau, \tag{5}$$

where $s_\theta(p_i^+)$ is the similarity metric of each pair of anchor-positive embeddings where $p_i^+ = (e, e_i^+)$ with $i \in \{1, ..., m^+\}$ and m^+ is the total number of positive examples, $s_\theta(p_j^-)$ indicates the metric of each pair of anchor-negative embeddings where $p_j^- = (e, e_j^-)$ with $i \in \{1, ..., m^-\}$ and m^- is the total number of negative examples, and τ is the minimum margin between those similarities.

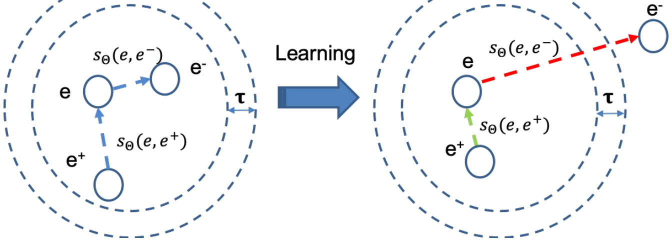


Fig. 2. Triplet loss learning process.

E. aAUC Loss

Metric learning loss functions aim to improve the generalization ability of the systems since they learn how to discriminate better for verification problems. However, these functions are designed without considering the measure of the performance used in the verification process. To make the training process more consistent with the evaluation procedure, in [27], we proposed to optimize directly the Area Under the Curve (*AUC*) as loss function using the triplet philosophy. AUC measures how much the model is capable of distinguishing between all the pairs of examples. This way, it provides a well-suited loss function that measures the performance of the whole system independently of the operating point. Therefore, AUC is a proper measure to make the training procedure consistent with the evaluation process. In order to enable the backpropagation of the gradients during the training process, we presented an effective differentiable approximation of AUC function (*aAUC*). Thus, given a set of network parameters θ, the *aAUC* loss function can be defined using a sigmoid function as,

$$aAUC(\theta) = \frac{1}{m^+m^-}\sum_{i=1}^{m^+}\sum_{j=1}^{m^-}\sigma_\alpha(s_\theta(p_i^+) - s_\theta(p_j^-)), \quad (6)$$

where $s_\theta(p_i^+)$ is the similarity metric of each pair of anchor-positive embeddings, $s_\theta(p_j^-)$ indicates the metric of each pair of anchor-negative embeddings, and $\sigma_\alpha()$ is the sigmoid function, expressed as,

$$\sigma_\alpha(s) = \frac{1}{1+exp(-\alpha \cdot s)}, \quad (7)$$

where α is an adjustable parameter set using development data. With this expression, the optimization process leads the score of the anchor-positive pair to be greater than the score of the negative pair.

Metric learning approaches have shown to be very effective techniques to improve the discrimination ability of SV systems. Especially this loss function based on the AUC SV metric, which is optimized in the learning framework to improve the system performance. However, these techniques with pairs or triplets have some drawbacks as slow convergence or instability. Traditionally, to solve these problems, sample mining strategies have been applied to select the most informative pairs to create the triplets [13]. This process improves the performance, but it also involves a high computational cost which slows down excessively the training process.

III. *aDCF* Loss Function

Motivated by the idea of taking advantage of the efficiency and speed of training using multi-class classification loss functions and, at the same time, the improvement achieved with verification loss functions as previous *aAUC* loss, we proposed the approximated Detection Cost Function (*aDCF*) [8]. This function is inspired by DCF [15] [16], which is one of the main performance measures in the evaluation process for SV tasks. Using *aDCF* loss function to train the systems, the network learns how to minimize this evaluation metric and find the optimal threshold for the specific application. In order to carry out this function, we had to develop an effective and differentiable expression for the original DCF metric.

aDCF loss function is composed of a weighted sum of the batch level estimate of the probability of misses or FRR (P_{miss}) and the probability of false alarm or FAR (P_{fa}). P_{miss} is defined by the average number of times the scores of target speakers N_{tar} are smaller than the decision threshold (Ω), so the system cannot effectively detect, and a miss is produced. While P_{fa} is determined by the average number of times the scores of non-target speakers N_{non} are greater than Ω, so a false alarm is produced. These two kinds of errors are graphically depicted in Fig.1 for each application case. Therefore, as a function of the network parameters θ, P_{fa} and P_{miss} can be written as,

$$P_{fa}(\theta,\Omega) = \frac{\sum_{y_i \in y_{non}} \mathbb{1}(s_\theta(x_i,y_i) > \Omega)}{N_{non}}, \quad (8)$$

$$P_{miss}(\theta,\Omega) = \frac{\sum_{y_i \in y_{tar}} \mathbb{1}(s_\theta(x_i,y_i) < \Omega)}{N_{tar}}, \quad (9)$$

where N_{tar} is the number of target speakers, N_{non} is the number of non-target speakers, $\mathbb{1}()$ is equal to *'1'* whenever the score $s_\theta(x_i,y_i)$ meets the condition with respect to Ω, and *'0'* otherwise. The score $s_\theta(x_i,y_i)$ is obtained from the last layer of the neural network where x_i is the input sample with $i \in \{1,...,m\}$ and m is the number of samples, y_i is the class label. Equations (8) and (9) can be rewritten using unit step function $u()$ as,

$$P_{fa}(\theta,\Omega) = \frac{\sum_{y_i \in y_{non}} u(s_\theta(x_i,y_i) - \Omega)}{N_{non}}, \quad (10)$$

$$P_{miss}(\theta,\Omega) = \frac{\sum_{y_i \in y_{tar}} u(\Omega - s_\theta(x_i,y_i))}{N_{tar}}. \quad (11)$$

However, the expressions (10) and (11) for the probabilities are not differentiable, so we replace unit step function $u()$ by a sigmoid of the difference to make an approximation of the binary counter which enables the backpropagation of gradients:

$$\hat{P}_{fa}(\theta,\Omega) = \frac{\sum_{y_i \in y_{non}} \sigma_\alpha(s_\theta(x_i,y_i) - \Omega)}{N_{non}}, \quad (12)$$

$$\hat{P}_{miss}(\theta,\Omega) = \frac{\sum_{y_i \in y_{tar}} \sigma_\alpha(\Omega - s_\theta(x_i,y_i))}{N_{tar}}. \quad (13)$$

Thus, using these expressions, we can now propose to minimize the following approximated loss function defined as,

$$aDCF(\theta,\Omega) = \gamma \cdot \hat{P}_{fa}(\theta,\Omega) + \beta \cdot \hat{P}_{miss}(\theta,\Omega), \quad (14)$$

where γ and β are configurable parameters that provide more cost relevance to one of the terms over the other. The effect of the values of these parameters with respect to the system application will be studied in the experimental section. For instance, some system applications provide more relevance when a target speaker is not detected, while other applications needs to decrease the number of non-target speakers that the system accepts. Note that Ω will be optimized as part of the system parameters. Moreover, to employ this function efficiently, we have adopted an implementation which follows the same training philosophy of the multi-class architectures with DNNs as we detail below.

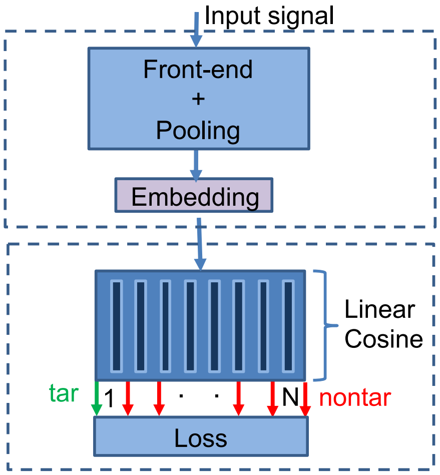


Fig. 3. Interpretation of the last layer of the neural network as a matrix of weights which models each identity.

A. Relationship between aDCF and CE Loss

To perform the multi-class classification task using DNNs, a widely adopted and efficient approach consists of training the system with CE loss combined with a softmax function. Thus, CE loss can be written as,

$$L_{CE} = -\frac{1}{m}\sum_{i}^{m} log \frac{\exp\left(s_\theta(x_i, y_i)\right)}{\sum_{j}^{N}\exp\left(s_\theta(x_i, j)\right)}, \qquad (15)$$

where x_i is the input sample with $i \in \{1, ..., m\}$ and m is the number of samples, y_i is the class label, N is the total number of classes, and $s_\theta(x_i, j)$ is obtained from the last layer of the DNN. Usually, the last layer is defined as a linear layer and the score for each class $s_\theta(x_i, y_i)$ is obtained as,

$$s_\theta(x_i, y_i) = W_{y_i}^T \cdot x_i + b_{y_i}, \qquad (16)$$

where x_i is the input of the last linear layer, $W_{y_i}^T$ is the row of the matrix of weights which contains the layer parameters of the speaker class y_i, and b_{y_i} is the bias term. Although in this work, we will remove the bias term to simulate score evaluations as the output of the last layer. The interpretation of the output of this layer as scores is possible since each vector of the matrix of weights used in the last layer can be interpreted as a model for each trained identity as Fig.3 depicts.

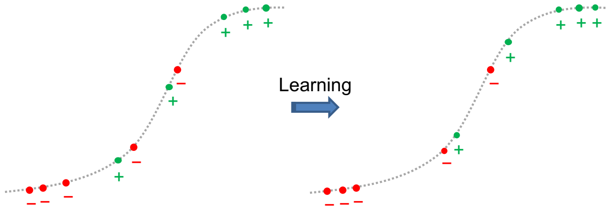


Fig. 4. aDCF learning process using a sigmoid function trained with target and non-target examples.

Despite its accuracy at the training process, CE loss is designed to maximize the posterior probability of the correct class in a multi-class classifier. Thus, this function is appropriate for classification tasks where the goal is to determine the identity of each sample in a known set of identities. However, in verification tasks, the main goal consists of detecting if two utterances belong to the same identity or not, so we need to measure the degree of separation and similarity. Therefore, the use of a sigmoid function as a binary classifier in aDCF loss function is more consistent with the speaker verification task and the evaluation process, since this strategy allows to construct one-versus-all comparison with multiple binary classifiers while the training efficiency is maintained. As in architecture with CE loss, the key to keep the efficiency for training the system with *aDCF* loss is the chance to interpret the matrix of weights as a representation of each trained speaker and obtain scores to optimize the system during the training progresses with it. Using this approach, the neural network is trained with positive samples versus negative samples to learn how to separate them with *aDCF* loss function based on the sigmoid function, see the example in Fig.4.

B. Efficient implementation

To develop this process efficiently, the training is made using subsets of samples, since standard neural network optimization operates with small batches of samples. When the network is trained with a batch size B, the number of N_{tar} and N_{non} that will be taken into account in expression (14) to calculate the gradients will be B, and $B(N-1)$, where N is the total number of speakers. Thus, as we can see in Fig.3, the target and non-target scores used in (12) and (13) are obtained using the ground truth labels and comparing with the model for each trained identity which is stored in the matrix of weights. Therefore, these assumptions allow us to obtain a similar efficiency and convergence speed to the most common approaches to multi-class classification as CE during the training process.

C. Cosine Distance Layer

Previous works [33] [34] [35] have remarked the fact that there is a gap between the metric used during training (16) and the cosine metric used in the evaluation. Also, these works have shown the relevance of the normalization of features and weights to provide significant performance improvements. Therefore, in this work, we have also analyzed the use of a cosine layer instead of a linear layer as the last layer in the neural network to obtain this score as,

$$s_\theta(x_i, y_i) = \frac{W_{y_i}^T \cdot x_i}{\left\|W_{y_i}^T\right\| \cdot \|x_i\|}, \qquad (17)$$

where $\|x_i\|$ is the normalized input signal to the last linear layer, and $\left\|W_{y_i}^T\right\|$ is the normalized layer parameters of the speaker class y_i. As we will show, the use of the cosine layer combined with aDCF loss function achieves better results.

IV. Speaker Verification System

Recent SV systems based on DNNs are trained for multi-class classification with a global average reduction mechanism

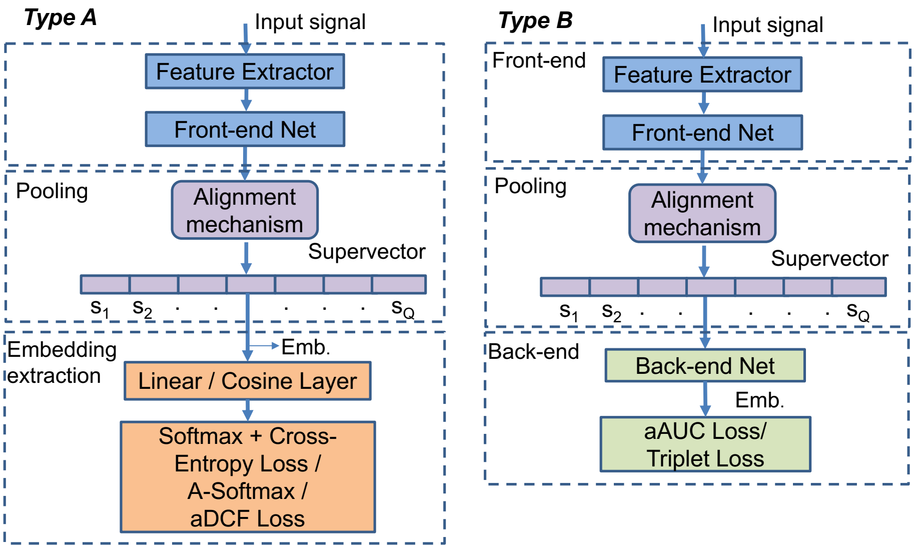


Fig. 5. Two architectures used to create the SV system. *Type A* is trained with two different loss functions, and in the case of Cross-Entropy loss, it is also used as pre-training for the other architecture. *Type B* is trained to optimize the back-end net for Triplet and aAUC loss.

which is applied to produce an embedding that represents each utterance. Once the system is trained, the verification process is performed through a back-end, either Probabilistic Linear Discriminant Analysis (PLDA) [36] [37], or a similarity metric [38] are predominant choices. Finally, verification results are rectified through score normalization (*snorm*) [39] and calibrated in order to choose an optimal threshold for detection. This last step ensures optimal performance for the operating point of the application.

In text-dependent tasks, this approach does not always work efficiently [9], because it dismisses the order of the phonetic information of the utterance to detect the correct speaker and phrase. In previous works [27] [40], we addressed this problem by replacing the average pooling mechanism by an alignment method as a new layer into the DNN architecture. This strategy keeps and encodes the temporal structure of the uttered phrase in a supervector. Fig.5 depicts the system structure employed for experiments, and each part of this system is described below.

A. Front-end

To address the issue of maintaining the temporal structure, we developed the front-end network of our system using Convolution layers of one dimension (1D convolution). These layers allow us to encode feature dynamics in the temporal dimension to add context information, and also, to combine all channels at each layer. This mechanism can be shown equivalent to TDNN layers. Furthermore, we employ the Bayesian Dark Knowledge [41] approach to model the uncertainty in the parameters during the training process. Using this approach, two copies of the network are trained simultaneously. The teacher network has to predict augmented unseen data, and the student network captures the variability in the predictions produced by the teacher network and encodes this uncertainty in the model, which produces a more robust system. Finally, the student network is used to extract the embeddings for the text-dependent task [42]. In [42], we showed how the use of this approach to train neural networks improves the system performance around 20%. Input to the front-end network are the acoustic features composed of the Mel-Frequency Cepstral Coefficients (MFCC) [43] [44] with their first and second derivatives.

B. Pooling

In many SV systems, the pooling process is made with a global average pooling mechanism which provides a fixed-length representation, also known as embedding. However, we introduced an alternative pooling mechanism based on a frame-to-components alignment process, which allows us to keep the order of the phonetic information of each phrase. In this work, the alignment mechanism used is a Gaussian Mixture Model (GMM) combined with a Maximum A Posteriori (MAP) adaptation [45].

C. Back-end

Front-end and pooling sections of the network are shared in both architectures employed. Nevertheless, the back-end part is different to make the verification process. Architecture Type A showed in Fig.5 combines the front-end and pooling with a cosine or linear layer and CE Loss, A-Softmax or *aDCF* loss to train the system. After the training process, verification scores are obtained applying a cosine similarity over embeddings.

On the other side, the architecture *type B* depicted in Fig.5 represents a trainable back-end with Triplet loss or *aAUC* loss as the objective function. As initialization for this architecture to avoid the convergence issues, a pre-trained model from the architecture *type A* with CE loss is employed.

V. Experimental Setup

A. Data

For the experimental results, we have employed the RSR2015 text-dependent speaker verification dataset [46].

TABLE I

EXPERIMENTAL RESULTS ON RSR2015-PARTI [46] EVAL SUBSET, SHOWING EER% AND NIST 2010 MIN AND ACTUAL COSTS (*minDCF*,*actDCF*). THESE RESULTS WERE OBTAINED BY SWEEPING OF THE PARAMETER VALUES IN aDCF LOSS FUNCTION WITH NORMALIZATION (SNORM).

aDCF Parameters			**Female**			**Male**			**Female+Male**		
γ	β	α	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**
0.15	0.85	1	1.22	0.328	0.646	1.19	0.279	0.611	1.21	0.312	0.322
		10	0.57	0.164	0.199	0.76	0.179	0.180	0.70	0.204	0.210
		20	0.45	0.086	0.163	**0.52**	0.132	0.134	0.52	0.137	0.148
		30	0.34	0.096	0.293	0.57	**0.110**	0.119	0.51	0.139	0.141
		40	0.43	0.095	0.117	0.57	0.119	0.121	0.55	0.124	0.132
0.25	0.75	1	1.18	0.300	0.772	0.90	0.245	0.248	1.04	0.272	0.516
		10	0.51	0.153	0.163	0.70	0.170	0.174	0.62	0.180	0.189
		20	0.44	0.085	0.118	0.67	0.140	0.141	0.60	0.133	0.133
		30	0.38	0.085	0.088	0.61	0.122	0.124	0.55	0.128	0.146
		40	0.43	0.094	0.210	0.62	0.116	0.133	0.59	0.126	0.206
0.50	0.50	1	1.15	0.311	0.587	0.91	0.267	0.270	1.05	0.300	0.330
		10	0.51	0.115	0.186	0.64	0.132	0.135	0.63	0.146	0.163
		20	0.35	0.093	0.093	0.54	0.129	0.143	0.51	0.119	0.128
		30	0.39	0.088	0.111	0.64	0.118	0.121	0.58	0.128	0.130
		40	0.36	0.072	0.104	0.63	0.120	0.138	0.56	**0.115**	0.123
0.75	0.25	1	3.81	0.723	0.984	6.83	0.797	1.000	6.59	0.797	0.999
		10	0.42	0.093	0.119	0.56	0.154	0.214	0.58	0.145	0.154
		20	0.44	0.094	0.097	0.56	0.125	0.158	0.55	0.131	0.134
		30	0.36	0.078	0.158	0.59	0.115	**0.118**	0.55	0.123	0.140
		40	0.33	**0.068**	0.084	0.55	0.116	0.150	**0.50**	0.117	**0.119**
0.85	0.15	1	4.69	0.709	0.986	7.14	0.794	0.874	6.79	0.781	0.999
		10	2.32	0.393	0.420	6.65	0.687	0.699	4.25	0.525	0.684
		20	0.34	0.071	**0.073**	0.88	0.187	0.445	0.68	0.157	0.159
		30	**0.32**	0.078	0.096	0.61	0.128	0.226	0.54	0.132	0.135
		40	0.38	0.077	0.080	0.59	0.125	0.225	0.57	0.118	0.120

This dataset comprises recordings from 157 male and 143 female, and 9 sessions for each speaker pronouncing 30 different phrases. Furthermore, this data is divided into three speaker subsets: background (bkg), development (dev) and evaluation (eval). In this work, we develop our experiments with Part I which contains 30 phonetically pass-phrases, and Part II which is based on 30 short control commands which have strong overlap of lexical content. These two parts have three subsets: background (bkg), development (dev), and evaluation (eval). In this paper, we employ the bkg (97 speakers, 47 female/50 male) for training and dev data (97 speakers, 47 female/50 male) for normalization and calibration. The eval set is used for enrollment and trial evaluation. This dataset has three evaluation conditions, but in this work, we have only evaluated the most challenging and employed in the text-dependent SV, which is the Impostor-Correct case. In this condition, non-target speakers pronounce the same phrase as the target speakers.

B. System configuration

In our experiments, we have used 20 dimension MFCCs stacked with their first and second order derivatives as input to train the alignment mechanism and as input to the DNN. Furthermore, a 64 component GMM has been trained per phrase using the bkg partition. From these models, the alignment information is extracted to use it in the alignment mechanism of each of the DNN architectures, since one model is trained for each different phrase. Furthermore, in this work, we have found effective the application of a phrase- and gender-dependent score normalization to conclude the system. We use a symmetric normalization, denoted *snorm* [39]. Afterwards, a calibration step is performed using linear logistic regression with the Bosaris toolkit [47].

VI. RESULTS

In this work, several sets of experiments have been developed with Part I and Part II. First, a set of experiments was carried out to study the behaviour of the system while different parameter values in *aDCF* loss function are swept. After that, we have analyzed the use of a complementary loss to improve the discrimination ability in combination with CE loss and *aDCF* loss with the second set of experiments. In the last set of experiments, we have evaluated the system employing some of the most extended state-of-the-art loss functions to compare the performance with our *aDCF* loss. Moreover, it should be noted that there are systems in the state-of-the-art for the RSR2015 dataset with relevant results, similar to those presented below. Nevertheless, such systems are based on traditional models such as Hidden Markov Models (HMMs) [46] [48] or neural network architectures focused on two different streams for speaker and utterance information [49] [50].

To evaluate the results of these experiments, we have measured the performance using *EER* [51] [52], NIST 2010 minimum and actual Detection Cost Function (*minDCF*, *actDCF*) [53].

A. aDCF Parameters α, γ, β

In this section, we analyze the system performance when we sweep *aDCF* loss parameters which are the terms of the cost

TABLE II

EXPERIMENTAL RESULTS ON RSR2015-PARTII [46] EVAL SUBSET, SHOWING EER% AND NIST 2010 MIN AND ACTUAL COSTS (*minDCF*,*actDCF*). THESE RESULTS WERE OBTAINED BY SWEEPING OF THE PARAMETER VALUES IN aDCF LOSS FUNCTION WITH NORMALIZATION (SNORM).

aDCF Parameters			**Female**			**Male**			**Female+Male**		
γ	β	α	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**
0.15	0.85	1	5.22	0.746	0.839	12.05	0.817	1.000	8.93	0.786	1.000
		10	3.24	0.588	0.602	7.17	0.721	0.771	5.42	0.666	0.744
		20	2.66	0.478	0.491	6.93	0.671	0.770	5.41	0.588	0.685
		30	2.59	0.434	0.446	4.54	0.615	0.626	3.75	0.537	0.543
		40	2.83	0.456	0.463	4.39	0.588	0.688	3.76	0.530	0.539
0.25	0.75	1	4.18	0.719	0.755	6.56	0.741	0.921	5.51	0.733	0.771
		10	2.84	0.502	0.541	4.96	0.624	0.626	4.10	0.566	0.570
		20	2.82	0.443	0.481	4.15	0.576	0.609	3.60	0.521	0.523
		30	2.67	0.428	0.446	**4.08**	0.583	0.587	3.50	0.514	0.516
		40	2.70	0.460	0.463	4.20	0.611	0.705	3.62	0.543	0.545
0.50	0.50	1	4.13	0.648	0.658	5.24	0.725	0.727	4.75	0.693	0.701
		10	2.76	0.489	0.518	**4.08**	0.606	0.628	3.51	0.551	0.575
		20	**2.47**	0.424	0.467	4.10	**0.568**	**0.570**	**3.42**	**0.506**	**0.515**
		30	2.54	0.425	**0.432**	4.21	0.596	0.598	3.54	0.518	0.528
		40	3.83	0.446	0.452	4.21	0.576	0.593	4.03	0.519	0.522
0.75	0.25	1	19.55	0.983	1.000	22.32	0.988	1.000	21.91	0.971	1.000
		10	2.84	0.486	0.529	4.95	0.633	0.645	4.05	0.566	0.613
		20	2.61	0.443	0.467	4.18	0.579	0.598	3.64	0.527	0.530
		30	2.57	0.437	0.448	4.17	**0.568**	**0.570**	3.56	0.514	0.520
		40	2.65	**0.422**	**0.432**	4.15	0.609	0.619	3.58	0.526	0.531
0.85	0.15	1	23.47	0.964	1.000	26.11	0.988	1.000	25.16	0.977	1.000
		10	18.08	0.940	0.995	21.69	0.974	1.000	20.14	0.968	1.000
		20	4.36	0.604	0.955	7.93	0.755	0.759	6.88	0.693	0.756
		30	3.29	0.475	0.680	5.92	0.697	0.705	4.99	0.595	0.653
		40	2.95	0.458	0.466	5.77	0.658	0.674	4.75	0.567	0.572

relevance (γ, β) and the adjustable parameter in the sigmoid function (α).

Table I shows *EER*, *minDCF* and *actDCF* results with Part I for different configurations of parameter values. As we can observe, in most of the experiments, the use of a greater value for α parameter improves results since α value modifies the slope of the sigmoid function. Thus, a greater value involves the sigmoid is closer to the unit step, which is used to define the exact *DCF* function. Furthermore, we observe that results are even better when we give more relevance to the probability of false alarms (P_{fa}) during the training with the cost term γ. However, we have noted that if this value is too extreme (upper 0.90-0.10), there are some convergence problems during the training process, and results are more sensitive to the variation of α value.

In addition to the previous table, Fig.6 depicts Detection Error Trade-off (DET) curves [54] which represents the relationship between P_{fa} (FAR) and P_{miss} (FRR). These curves show the best result for each configuration of the cost parameters. Note that these representations demonstrate the best results for each configuration of γ and β, and although the results are too similar, we decide to use as reference system the configuration with better behaviour in all the operating points which corresponds to $\gamma = 0.75$, $\beta = 0.25$ and $\alpha = 40$.

Results obtained for RSR2015-Part II are shown in Table II and Fig.7. In this set of experiments, phrases are shorter and have overlapped lexical content, so it is more challenging than Part I and the general performance is worst. In this case, we observe that when the cost terms are balanced, with an intermediate value of α is enough to achieve the best performance.

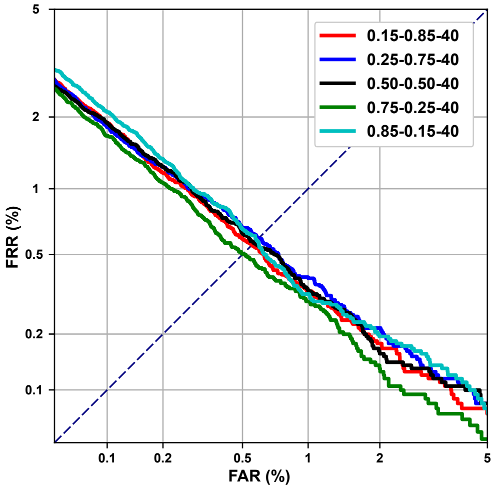


Fig. 6. DET curves for female+male results on RSR2015-Part I varying the parameter values of aDCF loss function γ and β, and using the best α value in each case.

For illustrative purposes, we have included Fig.8 to depict the evolution of *aDCF* with different α values against the exact DCF function during training. This figure shows that the differentiable approximation of the DCF function is getting close to the real function while the training progresses. Note that this evolution supports the fact that *aDCF* is an effective approximation of the real DCF.

B. Last Layer and Ring Loss Study

A second set of experiments was carried out to observe the system performance when a complementary loss as Ring loss

TABLE III

Experimental results on RSR2015-PartI [46] eval subset, showing EER% and NIST 2010 min and actual costs (*minDCF*,*actDCF*). These results were obtained to analyze the behaviour using a complementary loss with normalization (snorm).

Architecture			Female			Male			Female+Male		
Layer	**Loss**	**Ring**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**
Linear	CE	yes	**0.47**	**0.114**	**0.149**	**0.83**	**0.177**	**0.189**	**0.72**	**0.159**	**0.165**
		no	0.86	0.240	0.255	1.00	0.243	0.258	0.96	0.259	0.269
Cosine		yes	0.73	0.217	0.243	1.51	0.337	0.379	1.21	0.289	0.296
		no	0.92	0.282	0.375	0.91	0.245	0.288	0.94	0.264	0.314
Linear	aDCF	yes	1.39	0.320	0.764	2.23	0.487	0.939	2.09	0.420	0.768
		no	1.77	0.430	0.684	2.32	0.455	0.998	2.08	0.447	0.953
Cosine		yes	0.34	0.085	0.140	0.68	0.129	0.137	0.60	0.125	0.145
		no	**0.33**	**0.068**	**0.084**	**0.55**	**0.116**	**0.150**	**0.50**	**0.117**	**0.119**

TABLE IV

Experimental results on RSR2015-PartII [46] eval subset, showing EER% and NIST 2010 min and actual costs (*minDCF*,*actDCF*). These results were obtained to analyze the behaviour using a complementary loss with normalization (snorm).

Architecture			Female			Male			Female+Male		
Layer	**Loss**	**Ring**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**
Linear	CE	yes	3.31	**0.550**	**0.563**	5.45	0.690	0.833	4.58	**0.627**	**0.645**
		no	**3.20**	0.569	0.608	**5.03**	**0.689**	0.969	**4.25**	0.639	0.717
Cosine		yes	4.14	0.663	0.773	8.76	0.804	0.872	6.71	0.745	0.771
		no	3.68	0.641	0.647	5.24	0.719	**0.732**	4.55	0.689	0.718
Linear	aDCF	yes	8.69	0.823	0.832	12.57	0.883	0.890	10.97	0.857	0.874
		no	10.92	0.901	0.976	9.51	0.841	1.000	10.94	0.882	1.000
Cosine		yes	3.47	0.512	0.536	5.15	0.638	0.788	4.43	0.580	0.633
		no	**2.47**	**0.424**	**0.467**	**4.10**	**0.568**	**0.570**	**3.42**	**0.506**	**0.515**

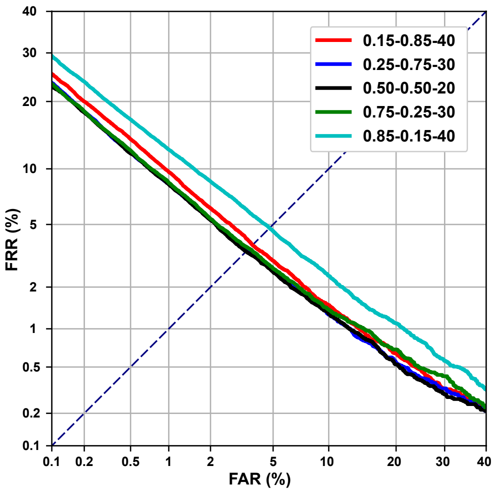


Fig. 7. DET curves for female+male results on RSR2015-Part II varying the parameter values of aDCF loss function γ and β, and using the best α value in each case.

is added to CE loss or *aDCF* loss. Results of these experiments in Part I (Table III) demonstrate that our *aDCF* loss function does not need a complementary loss function to improve the discrimination ability, while CE loss needs it. Furthermore, when we employ *aDCF* loss function to train the system, the use of a cosine layer as last layer is the most suitable option since this cosine metric used during the training process is the same metric that evaluation employs to obtain the final verification scores. Thus, we can see better results when the training process used is a pipeline more similar to the final verification process.

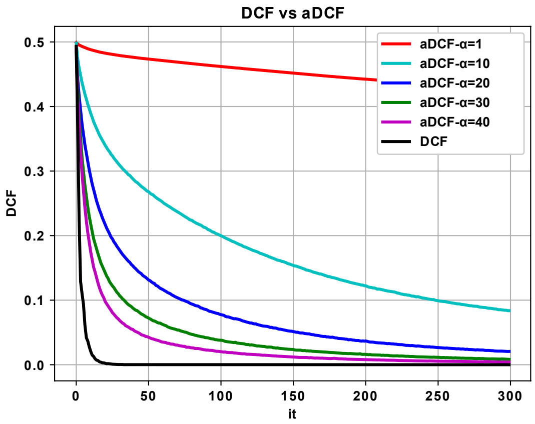


Fig. 8. Evolution training aDCF with different α values and exact DCF.

When these experiments are carried out for Part II, we observe in Table IV that the effect of using a complementary loss and a cosine layer instead a linear layer follows a similar trend as results of Part I. However, we have checked that the CE loss without Ring loss with this set of data provides similar results to the ones with Ring loss. The difficult in this set of data may cause that the system can not bring together correctly some features from the same identity even though Ring Loss is applied and for this reason, results are not so clearly improved when applying this complementary function.

C. Comparison with State-of-the-Art Loss Functions

In the last set of experiments, we have made a comparison among the best configurations of *aDCF* loss for Part I and Part

TABLE V
EXPERIMENTAL RESULTS ON RSR2015-PARTI [46] EVAL SUBSET, SHOWING EER% AND NIST 2010 MIN COSTS (*DCF10*). THESE RESULTS WERE OBTAINED TO COMPARE THE DIFFERENT LOSS FUNCTIONS WITH NORMALIZATION (SNORM).

Architecture			**Female**			**Male**			**Female+Male**	
Loss	**Type**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**
CE+RL	A	0.47	0.114	0.149	0.83	0.177	0.189	0.72	0.159	0.165
Trloss	B	0.78	0.161	0.559	0.96	0.174	0.319	0.88	0.173	0.261
aAUC	B	0.47	0.103	0.142	0.74	0.157	0.170	0.64	0.137	0.157
A-Softmax	A	0.68	0.156	0.263	0.70	0.163	0.187	0.70	0.178	0.187
aDCF	A	**0.33**	**0.068**	**0.084**	**0.55**	**0.116**	**0.150**	**0.50**	**0.117**	**0.119**

TABLE VI
EXPERIMENTAL RESULTS ON RSR2015-PARTII [46] EVAL SUBSET, SHOWING EER% AND NIST 2010 MIN COSTS (*DCF10*). THESE RESULTS WERE OBTAINED TO COMPARE THE DIFFERENT LOSS FUNCTIONS WITH NORMALIZATION (SNORM).

Architecture			**Female**			**Male**			**Female+Male**	
Loss	**Type**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**	**EER%**	**minDCF**	**actDCF**
CE+RL	A	3.31	0.550	0.563	5.45	0.690	0.833	4.58	0.627	0.645
Trloss	B	3.75	0.542	0.603	5.53	0.621	0.655	4.66	0.583	0.603
aAUC	B	2.76	0.503	0.527	4.62	0.601	0.760	4.25	0.579	0.704
A-Softmax	A	2.79	0.511	0.575	**4.01**	0.655	0.666	3.51	0.589	0.610
aDCF	A	**2.47**	**0.424**	**0.467**	4.10	**0.568**	**0.570**	**3.42**	**0.506**	**0.515**

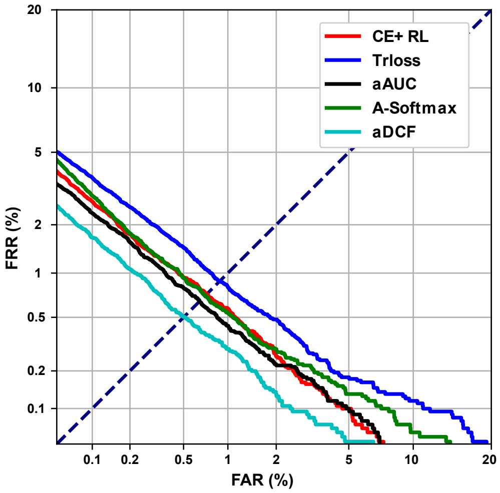


Fig. 9. DET curves for female+male results on RSR2015-Part I using different loss functions.

II and some of the most extended loss functions in the state-of-the-art which are CE loss combined with Ring loss (CE+RL), Triplet loss (Trloss), *aAUC* loss, and A-Softmax. In Table V, we show the results of these experiments with Part I. We can observe that architecture *type A* trained using *aDCF* loss function achieves the best results. Especially relevant is the improvement at comparing with CE loss and A-Softmax, since both have been trained using the most comparable strategy with architecture *type A*, which allows us to keep the same efficiency for the training process. In terms of relative improvement, EER% and minDCF values have improved 30.6% and 26.4% with respect to CE loss, and 28.6% and 34.3% with respect to A-Softmax. This improvement is remarkable given that the models involved have the same number of parameters and the inference time is not increased.

Moreover, we have also added Fig.9 with corresponding DET curves. These curves show results for female+male experiments. These representations demonstrate that the best system performance for all operating points is obtained for architecture *type A* trained with *aDCF*. On the other hand, note that in the literature A-Softmax was proposed as a better approach to replace the CE loss, but in this case, the system trained using the CE loss combined with Ring loss achieves a better overall performance.

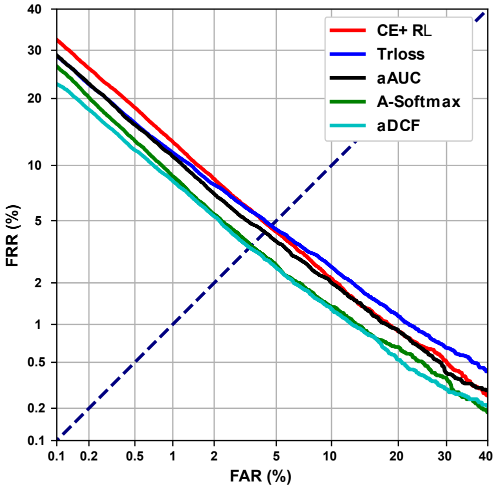


Fig. 10. DET curves for female+male results on RSR2015-Part II using different loss functions.

Results for these experiments using Part II are shown in Table VI and Fig.10. As in the experiments with Part I, the system trained with *aDCF* has the best results with a relative improvement in EER% and minDCF of 25.3% and 19.3% with respect to CE loss, and 2.6% and 14.1% compared to A-Softmax. In addition, it should be noted that the DET curves show that the system trained with *aDCF* loss obtains the best behaviour for low FAR operating points, while when FRR is low, the behaviour is similar to the system trained with A-Softmax.

D. Impact of the score normalization

In this section, we analyze the results without score normalization and calibration. To carry out this study, we only compare the same type of architectures: architecture $typeA$. Table VII obtained EER and minDCF results for systems with and without *snorm*. For systems with snorm, aDCF achieves the best performance in both Parts I and II. For systems without snorm at Part I, aDCF and A-Softmax present similar results without using score normalization. While the relative improvement in both metrics with respect to CE loss is higher than 50%. However, for the rows corresponding to Part II, A-Softmax outperforms aDCF. In the following, we further analyze the results without *snorm* by checking the performance at each phrase.

TABLE VII
RESULTS IN TERMS OF EER% AND NIST 2010 MIN COSTS (*DCF10*) FOR RSR2015-PARTI AND PARTII [46] EVAL SUBSET (FEMALE AND MALE) WITH AND WITHOUT NORMALIZATION (SNORM).

	Architecture		**Without SNORM**		**With SNORM**	
Dataset	**Loss**	**Type**	**EER%**	**minDCF**	**EER%**	**minDCF**
RSR-Part I	CE+RL	A	1.87	0.373	0.72	0.159
	A-Softmax	A	0.85	0.171	0.70	0.178
	aDCF	A	0.82	0.174	**0.50**	**0.117**
RSR-Part II	CE+RL	A	9.64	0.964	4.58	0.627
	A-Softmax	A	5.16	0.800	3.51	0.589
	aDCF	A	6.56	0.876	**3.42**	**0.506**

Table VIII shows the average improvement of aDCF against CE+RL/A-Softmax in terms of minDCF for results without normalization. Positive values indicate that aDCF obtains better performance regarding the other loss. This time we evaluated the phrases individually. Note that, at the phrase level, the system with an aDCF loss achieved the best performance for both parts of the dataset I and II. To explain this result, note that we trained a model by phrase, so with aDCF loss, the model learns to obtain the best score distributions for each phrase. Thus, when we separately evaluate the phrases, obtained scores have one optimal threshold by each. However, at joining together all scores (as in table VI), there is a single threshold for all of them. The application of a single decision threshold is non-optimal for all the phrases and this causes the drop in performance observed in Table VII.

TABLE VIII
AVERAGE IMPROVEMENT OF ADCF VS. CE+RL/A-SOFTMAX WITHOUT NORMALIZATION (SNORM) IN TERMS OF NIST 2010 MIN COSTS (*DCF10*) BY PHRASE ON RSR2015-PARTI AND PARTII [46] EVAL SUBSET.

	minDCF(%Improv.)	
	RSR-Part I	**RSR-Part II**
aDCF vs CE+RL	9.39	14.85
aDCF vs A-Softmax	13.41	7.36

VII. CONCLUSION

In this paper, we have made a wide analysis of the use of a metric learning approach based on aDCF loss function. This function is an approximated measurement of the decision errors FAR and FRR in SV systems which allows end-to-end systems to optimize a metric used in the final verification process. To employ this loss function, we have used an efficient implementation which follows the philosophy of existing multi-class loss functions and allows us to take advantage of the interpretation of the matrix of weights of the last layer to obtain directly the scores as the training progress.

Experiments to evaluate the effectiveness of our approach are carried out in RSR2015-Part I and Part II text-dependent speaker verification database. Results obtained studying the swept of aDCF loss parameters have shown great performance in both datasets, and also that in function of the part of the database, the best parameter configuration is different. Furthermore, we have checked the improvement achieved when aDCF loss function is combined with a cosine distance layer as last layer in DNN instead of the usual linear layer. It has been also observed that aDCF loss does not need the use of a complementary loss to improve the discrimination ability, while CE loss improves considerably with it. Finally, we compared results obtained using aDCF loss with some of the state-of-the-art approaches, and aDCF loss outperforms all of them with relative improvements upper 10% in the EER and DCF metrics.

ACKNOWLEDGMENT

This work has been supported by the European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie Grant 101007666; in part by MCIN/AEI/10.13039/501100011033 and by the European Union "NextGenerationEU" / PRTR under Grant PDC2021-120846-C41, by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the grant PRE2018-083312, by the Government of Aragón (Grant Group T36_20R), and by Nuance Communications, Inc.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[2] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on circuits and systems for video technology*, vol. 14, no. 1, pp. 4–20, 2004.

[3] J. Gonzalez-Rodriguez, "Evaluating automatic speaker recognition systems: An overview of the nist speaker recognition evaluations (1996-2014)," *Loquens*, 2014.

[4] D. A. Van Leeuwen and N. Brümmer, "An introduction to application-independent evaluation of speaker recognition systems," in *Speaker classification I*. Springer, 2007, pp. 330–353.

[5] S. Z. Li, *Encyclopedia of Biometrics*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[6] R. Togneri and D. Pullella, "An overview of speaker identification: Accuracy and robustness issues," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 23–61, 2011.

[7] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *speech communication*, vol. 66, pp. 130–153, 2015.

[8] V. Mingote, A. Miguel, D. Ribas, A. Ortega, and E. Lleida, "Optimization of False Acceptance/Rejection Rates and Decision Threshold for End-to-End Text-Dependent Speaker Verification Systems," *Proc. Interspeech 2019*, pp. 2903–2907, 2019.

[9] E. Malykh, S. Novoselov, and O. Kudashev, "On residual CNN in text-dependent speaker verification task," in *International Conference on Speech and Computer*. Springer, 2017, pp. 593–601.

[10] 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)*. IEEE, 2018, pp. 5329–5333.

[11] A. Krizhevsky, I. Sulskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information and Processing Systems (NIPS)*, pp. 1–9, 2012.

[12] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. Interspeech*, 2017, pp. 1517–1521.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[14] C. Zhang, K. Koishida, and J. H. Hansen, "Text-independent speaker verification based on triplet convolutional neural network embeddings," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 9, pp. 1633–1644, 2018.

[15] A. Martin and M. Przybocki, "The NIST 1999 speaker recognition evaluation—An overview," *Digital signal processing*, vol. 10, no. 1-3, pp. 1–18, 2000.

[16] S. Bengio and J. Mariéthoz, "The expected performance curve: a new assessment measure for person authentication," IDIAP, Tech. Rep., 2003.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[18] Y. Srivastava, V. Murali, and S. R. Dubey, "A Performance Comparison of Loss Functions for Deep Face Recognition," *arXiv preprint arXiv:1901.05903*, 2019.

[19] Y. Li, F. Gao, Z. Ou, and J. Sun, "Angular Softmax Loss for End-to-end Speaker Verification," *arXiv preprint arXiv:1806.03464*, 2018.

[20] I. Kukanov, T. N. Trong, V. Hautamäki, S. M. Siniscalchi, V. M. Salerno, and K. A. Lee, "Maximal Figure-of-Merit Framework to Detect Multi-Label Phonetic Features for Spoken Language Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 682–695, 2020.

[21] Y. Zheng, D. K. Pal, and M. Savvides, "Ring loss: Convex feature normalization for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5089–5097.

[22] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[23] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[24] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[25] Z. Bai, X. Zhang, and J. Chen, "Partial AUC Optimization Based Deep Speaker Embeddings with Class-Center Learning for Text-Independent Speaker Verification," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6819–6823.

[26] S. Ramoji, P. Krishnan, and S. Ganapathy, "NPLDA: A Deep Neural PLDA Model for Speaker Verification," *arXiv preprint arXiv:2002.03562*, 2020.

[27] V. Mingote, A. Miguel, A. Ortega, and E. Lleida, "Optimization of the area under the roc curve using neural network supervectors for text-dependent speaker verification," *Computer Speech & Language*, vol. 63, p. 101078, 2020.

[28] V. Mingote, D. Castan, M. McLaren, M. K. Nandwana, A. Ortega, E. Lleida, and A. Miguel, "Language Recognition Using Triplet Neural Networks," *Proc. Interspeech 2019*, pp. 4025–4029, 2019.

[29] L. P. Garcia-Perera, J. A. Nolazco-Flores, B. Raj, and R. Stern, "Optimization of the DET curve in speaker verification," in *Spoken Language Technology Workshop (SLT), 2012 IEEE*. IEEE, 2012, pp. 318–323.

[30] K.-A. Toh, J. Kim, and S. Lee, "Maximizing area under ROC curve for biometric scores fusion," *Pattern Recognition*, vol. 41, no. 11, pp. 3373–3392, 2008.

[31] A. Herschtal and B. Raskutti, "Optimising area under the ROC curve using gradient descent," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 49.

[32] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[33] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained Softmax Loss for Discriminative Face Verification," *arXiv preprint arXiv:1703.09507*, 2017.

[34] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille, "Normface: L2 hypersphere embedding for face verification," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1041–1049.

[35] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[36] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

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

[38] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.

[39] N. Brümmer and A. Strasheim, "Agnitio's speaker recognition system for evalita 2009," in *The 11th Conference of the Italian Association for Artificial Intelligence*. Citeseer, 2009.

[40] V. Mingote, A. Miguel, A. Ortega, and E. Lleida, "Supervector Extraction for Encoding Speaker and Phrase Information with Neural Networks for Text-Dependent Speaker Verification," *Applied Sciences*, vol. 9, no. 16, p. 3295, 2019.

[41] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *Journal of Animal and Plant Sciences*, vol. 27, no. 3, pp. 797–802, 2015.

[42] V. Mingote, A. Miguel, D. Ribas, A. Ortega, and E. Lleida, "Knowledge Distillation and Random Erasing Data Augmentation for Text-Dependent Speaker Verification," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6824–6828.

[43] P. Mermelstein, "Distance measures for speech recognition, psychological and instrumental," *Pattern recognition and artificial intelligence*, vol. 116, pp. 374–388, 1976.

[44] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE transactions on acoustics, speech, and signal processing*, vol. 28, no. 4, pp. 357–366, 1980.

[45] D. A. Reynolds, R. C. Rose *et al.*, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE transactions on speech and audio processing*, vol. 3, no. 1, pp. 72–83, 1995.

[46] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014.

[47] N. Brümmer and E. De Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

[48] R. K. Das, M. Madhavi, and H. Li, "Compensating utterance information in fixed phrase speaker verification," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1708–1712.

[49] T. Liu, M. C. Madhavi, R. K. Das, and H. Li, "A Unified Framework for Speaker and Utterance Verification." *Proc. Interspeech 2019*, pp. 4320–4324, 2019.

[50] T. Liu, R. K. Das, M. Madhavi, S. Shen, and H. Li, "Speaker-Utterance Dual Attention for Speaker and Utterance Verification," *Proc. Interspeech 2020*, pp. 4293–4297, 2020.

[51] J. M. Bernardo and A. F. Smith, *Bayesian theory*. John Wiley & Sons, 2009, vol. 405.

[52] N. Brümmer and J. Du Preez, "Application-independent evaluation of speaker detection," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 230–275, 2006.

[53] "The NIST Year 2010 Speaker Recognition Evaluation Plan," 2010. [Online]. Available: https://www.nist.gov/sites/default/files/documents/itl/iad/mig/NIST_SRE10_evalplan-r6.pdf

[54] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," National Inst of Standards and Technology Gaithersburg MD, Tech. Rep., 1997.

Victoria Mingote is a Ph.D. candidate at the University of Zaragoza, Spain. She has achieved several publications of her work in different international journals and conference proceedings. Her research interests expands through the areas of signal processing, machine learning, multimodal verification and identification (voice and face), and language identification. She received the Bachelor's and Master's degree in Telecommunication Engineering from the University of Zaragoza, Spain, in 2014 and 2016, respectively.

Antonio Miguel was born in Zaragoza, Spain. He received the M.Sc. degree in telecommunication engineering and the Ph.D. degree from the University of Zaragoza (UZ), Zaragoza, Spain, in 2001 and 2008, respectively. From 2000 to 2006, he was with the Communication Technologies Group, Department of Electronic Engineering and Communications, UZ, under a research grant. Since 2006, he has been an Associate Professor in the Department of Electronic Engineering and Communications, UZ. His current research interests include acoustic modeling for speech and speaker recognition.

Dayana Ribas is a scientific researcher in the field of machine learning and statistical data analysis applied to speech signals. Received the Ph.D. degree in 2016 from the Technological University of Havana, in the framework of international research collaboration actions with Zaragoza University, Spain, and INRIA, France. Since 2018 she is research scientist with VIVOLAB, University of Zaragoza. Her research experience expands through the robust speech processing field, focusing mainly on robust speech processing, speaker recognition in noisy environments, realism in speech and language processing, and speech quality measures.

Alfonso Ortega received the Telecommunication Engineering and the Ph.D. degrees from the University of Zaragoza, Zaragoza, Spain, in 2000 and 2005, respectively. He is Associate Director of the Aragon Institute for Engineering Research (I3A), University of Zaragoza. In 2006, he was visiting scholar in the Center for Robust Speech Systems, University of Texas at Dallas, USA. He has participated in more than 50 research projects funded by national or international public institutions and more than 30 research projects for several companies. He is the author of more than 100 papers published in international journals or conference proceedings and several international patents. He is currently Associate Professor in the Department of Electronic Engineering and Communications, University of Zaragoza. His research interests include speech processing, analysis and modeling, automatic speaker verification, and automatic speech recognition. His Ph.D. thesis, advised by Dr. E. Lleida, received the Ph.D. Extraordinary Award and the Telefonica Chair Award to the best technological Ph.D.

Eduardo Lleida received the M.Sc. degree in telecommunication engineering and the Ph.D. degree in signal processing from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain, in 1985 and 1990, respectively. From 1986 to 1988, he was involved in his doctoral work in the Department of Signal Theory and Communications, UPC. From 1989 to 1990, he was an Assistant Professor, and from 1991 to 1993, he was an Associated Professor in the Department of Signal Theory and Communications, UPC. From February 1995 to January 1996, he was with ATT Bell Laboratories, Murray Hill, NJ, USA, as a Consultant in speech recognition. He is currently a full Professor of signal theory and communications in the Department of Electronic Engineering and Communications, University of Zaragoza, Zaragoza, Spain, and a member of the Aragon Institute for Engineering Research, where he is heading the ViVoLab research group in speech technologies. He has been the doctoral advisor of 12 doctoral students. He has managed more than 50 speech-related projects, being inventor in seven worldwide patents, and coauthored more than 200 technical papers in the field of speech, speaker and language recognition, speech enhancement and recognition in adverse acoustic environments, acoustic modeling, confidence measures, and spoken dialogue systems.