Adversarial Reweighting for Speaker Verification Fairness

Minho Jin1, Chelsea J.-T. Ju2, Zeya Chen2, Yi-Chieh Liu2, Jasha Droppo2, and Andreas Stolcke2

1Amazon Web Services, Palo Alto, CA, USA
2Amazon Alexa AI, Sunnyvale, CA, USA
{minhoj,juitij,zeyachen,ayliua,drojasha,stolcke}@amazon.com

Abstract

We address performance fairness for speaker verification using the adversarial reweighting (ARW) method. ARW is reformu-
alized for speaker verification with metric learning, and shown to improve results across different subgroups of gender and na-
tionality, without requiring annotation of subgroups in the train-
ing data. An adversarial network learns a weight for each train-
ing sample in the batch so that the main learner is forced to focus on poorly performing instances. Using a min-max opti-
mization algorithm, this method improves overall speaker veri-
fication fairness. We present three different ARW formulations: accumulated pairwise similarity, pseudo-labeling, and pairwise
weighting, and measure their performance in terms of equal
error rate (EER) on the VoxCeleb corpus. Results show that the pairwise weighting method can achieve 1.08% overall EER, 1.25% for male and 0.67% for female speakers, with relative
EER reductions of 7.7%, 10.1% and 3.0%, respectively. For
nationality subgroups, the proposed algorithm showed 1.04%
EER for US speakers, 0.76% for UK speakers, and 1.22% for all
others. The absolute EER gap between gender groups was re-
cduced from 0.70% to 0.58%, while the standard deviation over
nationality groups decreased from 0.21 to 0.19.

Index Terms: speaker verification, speaker recognition, fair-
ness.

1. Introduction

Speaker verification means to determine whether an input
speech utterance matches an enrolled speaker [1, 2]. This is a
key machine learning (ML) problem for voice assistants, as
it enables the system to provide a personalized user experience.
Given the broad acceptance of voice assistants (such as Ama-
zon Alexa, Google Home, and Apple Siri) in everyday life, we
consider it important that systems perform uniformly well for
all user groups. Problems with performance fairness for speech
processing systems have been pointed out with respect to race
[3] and nonnative accents [4]. Here we address performance
fairness specifically for speaker recognition, and specifically,
speaker verification, where fairness has not received much study
so far.

One approach to avoid performance disparities for groups
of speakers is to sample more data from underrepresented
groups [5, 6, 7, 8, 9]. Sampling from underrepresented groups
is based on assumptions about their distribution. If those as-
sumptions are incorrect, this can create its own biases relative to
attributes such as gender or nationality. Also, obtaining the re-
quired annotations may be difficult due to privacy concerns [5].
To address the fairness problem, many researches have been fo-
cusing on identifying the biases in the training data [5, 6]. These
include applications to face recognition [10, 11], image classifi-
cation [12], speech recognition [13], natural language process-
ing [14, 15], and classification [16, 17]. For speaker verifica-
tion, Fenu et al. [18] showed how data balancing in training can
improve fairness. Toussaint and Ding [19] demonstrated that
an end-to-end ResNet speaker verification model suffers from
performance divergence for different nationalities in VoxCeleb
data. Shen et al. [20] recently showed that using fusion net-
works trained on subgroups (e.g., male and female speakers)
can improve speaker verification fairness, including for unbal-
anced training data.

In this work, we focus on approaches that mitigate uneven
performance for different data subsets without using additional
attributes (such as gender and nationality) in the data, motivated
by Laotli et al. [21]. Similarly to the distributionally robust opti-
mization (DRO) method proposed by Hashimoto et al. [22],
this approach aims to automatically identify the underperfor-
mance groups using an adversarial network, and adjusts their con-
tribution to the training loss. This was shown to not only reduce
the performance disparity for minority groups, but to also im-
prove performance overall. The adversarial network is trained
so that it can identify and boost the weight of underperforming
data using a min-max optimization [21]:

\[
J(\theta, \phi) = \min_{\theta} \max_{\phi} \sum_{(x_i, y_i) \in D \times L} \lambda_{\phi}(x_i, y_i) \ell(h_{\theta}(x_i), y_i)
\]

(1)

where D and L are data and label sets, respectively. Given a
data sample xi with its label yi, the loss is computed as \ell(h_{\theta}(x_i), y_i) using the learner network parameters \theta. The
adversarial weight \lambda_{\phi}(x_i, y_i) is generated by the adversarial net-
work with parameters \phi. Given a learner network parameter \theta,
the adversary parameters \phi are first updated to maximize the
weighted loss J(\theta, \phi), which is expected to increase \lambda_{\phi}(x_i, y_i)
for underperforming groups. Then, J(\theta, \phi) is minimized with
respect to \theta to reduce the training loss.

We address how to apply this min-max optimization for
speaker verification with metric learning, more specifically with
angular prototypical loss [23, 24]. Unlike the classification
problem in (1), where \ell(h_{\theta}(x_i), y_i) is determined by the data
instance and its label, the loss for metric learning is defined us-
ing a batch of utterances from multiple speakers, and the ability
of the learner to classify pairs of utterances. For this purpose
we propose three different formulations: accumulated pairwise
similarity (APS), pseudo-labeling (PL) with K-means, and pair-
wise weighting (PW). In APS, the adversarial network com-
putes each speaker’s weight using the aggregated similarity to
other speakers. In PL, each speaker is labeled in an unsup-
vised manner using K-means, and the adversarial weights are a
function of these pseudo-labels. Finally, in PW, we make the
adversarial weights a function of pairs of speakers. We evalu-
ate the various approaches in terms of overall and group-wise
equal error rate (EER) for speaker verification on the VoxCeleb
corpus.

Section 2 describes the baseline speaker verification sys-
tem and the proposed ARW algorithms for speaker verification.
2. Algorithms

2.1. Speaker verification

The baseline speaker verification in our experiments uses utterance embeddings based on a ResNet model [24, 25, 26]. Given an audio input, the system extracts an embedding by feeding the Mel-band spectrogram of the audio to the ResNet model. Given a pair of embeddings, it computes their cosine values to make a decision on whether they come from the same speaker.

When training the speaker verification model, we used angular prototypical (AP) and label loss [23, 24]. An input batch for training consists of \( N \) speakers. For AP loss, we compute the anchor \( x_j^a \) and the query \( x_j^q \) for speaker \( j \):

\[
x_j^a = \frac{1}{M-1} \sum_{m=0}^{M-2} x_{j,m},
\]

\[
x_j^q = x_{j,M-1},
\]

(2)

(3)

where \( x_{j,m} \) for \( m \in \{0, \ldots, M-1\} \) denotes the embedding extracted from the \( m \)th utterance from the speaker \( j \in \{0, \ldots, N-1\} \). Using the anchor and the query, the similarity between a pair of speakers \( (j,k) \) is computed as follows:

\[
S_{j,k} = w \cos(x_j^a, x_k^q) + b,
\]

(4)

where

\[
\cos(x_j^a, x_k^q) = \frac{x_j^a \cdot x_k^q}{||x_j^a|| \cdot ||x_k^q||}
\]

(5)

and where \( w \) and \( b \) are trainable weight and bias coefficients, respectively. Finally the AP loss is computed according to

\[
L_p = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} L_{p,j,k},
\]

(6)

where

\[
L_{p,j,k} = -\log \frac{e^{S_{j,k}}}{\sum_{k=0}^{N-1} e^{S_{j,k}}}
\]

(7)

For the label loss, all utterances in the training set are labeled by their speaker index in the training data. The embeddings are fed into a single-layer network whose output size is the same as the number of training speakers, and the loss is defined as the softmax loss given the true speaker index. For this paper, we have adopted the best-performing model from [27], which uses a combination of AP and label loss.

2.2. Adversarial re-weighting algorithm

Fig. 1 illustrates how the representation bias of the training data affects fairness when classifying data into two classes, marked as circles and crosses. There are groups of red, blue and green data points, where green represents the minority group. The top of Fig. 1 is an example of the optimal classification boundary for the entire training set. However, the minority group (green) has 100% classification error. If we can identify this green group and increase their contribution to the overall training loss, it may be possible to obtain a classification boundary as shown at the bottom of the figure. The two classification boundaries will give the same number of classification errors, marked as dotted circles, but the decision boundary at the bottom provides better fairness at the group level.

In ARW, the adversarial network is trained along with the learner network so that it can learn to implicitly identify under-performing groups using the adversarial network. Fig. 2 shows
the elements of the ARW algorithm for a classification task. Given input data $x_i$ and label $y_i$, the learner makes a prediction of the class $y_i$. In [21], the weight $f_\phi(x_i, y_i)$ is computed using a sigmoid for each class of $Y$ so that it ranges over $[0, 1]$, where $\phi$ is the set of parameters of the adversary network. Finally, the weight $\lambda_\phi(x_i, y_i)$ is normalized:

$$\lambda_\phi(x_i, y_i) = 1 + \frac{f_\phi(x_i, y_i)}{B \sum_{j=0}^{B-1} f_\phi(x_j, y_j)},$$  

(8)

where $B$ is the batch size. The training is done as in (1), where the adversary network $\phi$ tries to maximize the loss (i.e., up-weighting the worst inputs or groups) and the learner (parameterized by $\theta$) tries to minimize the loss. In the implementation, we first train the learner for a sufficient number of epochs, and then iterate maximization of $J(\theta, \phi)$ with respect to $\phi$ and minimization of $J(\theta, \phi)$ with respect to $\theta$. In this original formulation, the adversary model output $f_\phi(x_i, y_i)$ is defined for the case of single-input classification, and thus needs to be adapted for metric learning, as in speaker verification.

Fig. 3 illustrates the reformulated ARW framework for speaker verification. Note that we feed the output embedding coming from the learner, rather than its input audio, to the adversary network. This simplifies the adversary network, since it takes in a fixed-length vector rather than a variable-length sequence of vectors. Within this general new framework, approaches differ in how the adversarial weights are utilized in the loss computation, as described next.

2.2.1. Accumulated pairwise similarity (APS)

Fig. 4 illustrates how we compute weights in the APS formulation. Unlike the original ARW, the adversary network $f_\phi$ computes a vector instead of a scalar, mapping the anchor vectors to a new space. Subsequently, the adversarial weights are computed from a comparison of the anchor to all speakers, using the inner product:

$$J(\theta, \phi) = \min_\theta \max_\phi \sum_{x_j, k \in D} \left( \sum_{j=0}^{N-1} \lambda_\phi(j, \{\forall k x_k^a\}) L_{p,j} \right).$$  

(9)

We formulate two versions of APS adversarial weights; one based on inner product similarity:

$$\lambda_\phi(j, \{\forall k x_k^a\}) = 1 + \frac{\sum_{k \in D} f_\phi(x_j^a) \cdot f_\phi(x_k^a)}{\sum_{k \in D} f_\phi(x_k^a) \cdot f_\phi(x_k^a)},$$  

(10)

and one based on cosine similarity:

$$\lambda_\phi(j, \{\forall k x_k^a\}) = 1 + \frac{\sum_{k \in D} e^{\cos(f_\phi(x_j^a), f_\phi(x_k^a))}}{\sum_{k \in D} \sum_{k' \in D} e^{\cos(f_\phi(x_j^a), f_\phi(x_{k'}^a))}}.$$  

(11)

Here $f_\phi(x_k^a) \in \mathbb{R}^H$ is the adversarial network, and $H$ is its output embedding dimension, a hyperparameter. We use the exponential of cosine similarity in (11) with an assumption that it is normally distributed.

Fig. 5 illustrates how we compute weights with PL. In this approach, we use the K-means algorithm [28] to group speakers based on their embeddings, and label each speaker as one of $K$ clusters:

$$\hat{y}_j = \arg\max_{k \in \{0, \ldots, K-1\}} \cos(c_k, x_j^a),$$  

(12)

where $c_k$ is the $k$th centroid. The learner is trained to output similar embeddings for similar speakers; thus similar speakers tend to end up with the same cluster index $\hat{y}_j$. We can consider the cluster index as a pseudo-speaker label, and have it take the place of the label $y_j$ in the standard ARW adversary formulation.

The adversary network $f_\phi$ is built as a multilayer perceptron network with nonlinear activation. The input is the embedding extracted from the anchor utterance, and the output layer has the probabilities of the $K$ classes, $f_\phi(x_j^a, \hat{y}_j)$ for $\hat{y}_j \in \{0, \ldots, K-1\}$. This leads to

$$J(\theta, \phi) = \min_\theta \max_\phi \sum_{x_j, k \in D} \left( \sum_{j=0}^{N-1} \lambda^K_{\phi}(x_j^a, \hat{y}_j) L_{p,j} \right),$$  

(13)

where the adversarial weight is computed by normalizing the adversary network outputs:

$$\lambda^K_{\phi}(x_j^a, \hat{y}_j) = 1 + \frac{f_\phi(x_j^a, \hat{y}_j)}{\sum_{k=0}^{K-1} f_\phi(x_k^a, \hat{y}_j)}.$$  

(14)

Unlike in APS, in PL the adversary network has $K$ output nodes, whereas $f_\phi(x_j^a, \hat{y}_j) \in \mathbb{R}$ as in the original ARW.

2.2.2. Pseudo-labeling with K-means (PL)

Unlike in the previous two versions of ARW, and as shown in Fig. 6, we formulate two versions of the adversarial weight as a
function of speaker pairs, to match the structure of AP loss:

$$J(\theta, \phi) = \min_{\phi} \max_{\theta} \sum_{x, y \in [D]} \left( -\frac{1}{N} \sum_{j=0}^{N-1} \log \frac{e^{\lambda_\phi(x_j^a, y_j^a)S_{j,a}}}{\sum_{k=1}^{N} e^{\lambda_\phi(x_j^a, y_k^a)S_{j,a}}} \right), \quad (15)$$

and

$$J(\theta, \phi) = \min_{\phi} \max_{\theta} \sum_{x, y \in [D]} \left( -\frac{1}{N} \sum_{j=0}^{N-1} \lambda_\phi(x_j^a, y_j^a)S_{j,a} \right). \quad (16)$$

The first version (15) applies the adversarial weight to the similarities between speakers while the second (16) applies it to the exponential of the similarities. We decompose $$\lambda_\phi(x_j^a, y_j^a)$$ as a combination of the single-speaker PL weights (14) to avoid doubling the input dimensionality:

$$\lambda_\phi(x_j^a, y_j^a) = \lambda^K_\phi(x_j^a, y_j^a) + \lambda^K_\phi(x_k^a, y_k^a), \quad (17)$$

where $$\hat{y}_j$$ and $$\tilde{y}_j$$ are the centroid indices computed using (12).

3. Experiments

3.1. Experimental setup

The performance of ARW was evaluated on the VoxCeleb1 database [29], with models trained on VoxCeleb2 data [24]. The training set comprised 5,994 speakers and 1,092,009 utterances. For training, we form batches of two utterances each from 200 speakers: $$N = 200$$ and $$M = 2$$. The backbone model was half ResNet34 from VoxCelebTrainer, with 8M parameters, 4 residual layers and attentive statistical pooling [25, 26, 30]. The learning rate was 0.001, and for every 3 epochs, the learning rate was decayed by a factor of 0.95. The evaluation was performed following the protocol from [24]. The speaker embedding size was 512, the default for VoxCelebTrainer.

3.2. Results

Table 1 enumerates the overall and the gender-group EERs of the baseline model, APS, PL, and PW. For APS, we varied the adversarial network model size from one up to three layers, and layer size (including the output dimension $$H$$) from 32 to 128 hidden units. Overall EER ranged from 1.09% to 1.15% for APS (10) and from 1.12% to 1.16% for APS (11); the best result was obtained with one hidden layer of 64 units and 128 units, respectively. For PL and PW, K-means clustering is performed at the beginning of each epoch with $$K = 8, 32$$, and 128. The adversarial network is a 256x3 multilayer perceptron network with sigmoid activation. In both cases, $$K = 128$$ achieved the lowest EER, namely, 1.08% for PL and 1.08% for PW: $$K = 8$$ and 32 leads to 1.10% and 1.13% for PL, and 1.12% and 1.11% for PW, respectively.

When comparing APS versions, we observe that the exponential of cosines (11) yielded a smaller gap between groups than inner products (10), though the latter had lower overall error. The EER difference between the baseline and PL is 0.06% for female and 0.12% for male speakers. The improvement for male speakers, which have higher EER, was larger than for female speakers. The EER gap between male and female speakers decreased from 0.70% to 0.58% with PW. When comparing PW (15) and PW (16), we observe that the latter more effectively reduced the EER gap between gender groups.

Table 2 enumerates the nationality-dependent EERs. In APS (10), we observe that the standard deviation remains similar to the baseline, while APS (11) reduced the standard deviation to 0.19. We observe that the EER difference between the baseline and PL is 0.02% for US, 0.03% for UK, and 0.06% for other speakers. The standard deviation of EERs decreased from 0.21 to 0.20. Also with PW, the standard deviation is further reduced to 0.19. The gain for the majority group (US speakers) is modest, and the gain in overall performance is driven by the minority groups (UK and Others). The improvement based on nationality breakdown is not as clear as that based on gender. We conjecture that nationality is a more complex and ambiguous attribute than gender, with a distribution that is highly unbalanced (mostly US and UK). Also, the Others class lump together many subgroups with diverse characteristics, and US and UK populations are themselves diverse with substantial overlaps in terms of speech characteristics, making gains harder to detect via our three-way breakdown of the speaker population.

4. Conclusions

We have proposed a novel approach to speaker verification fairness based on adversarial reweighting. We compared different reweighting algorithms based on adversarial networks, and reduced the EER gap between different groups based on gender and nationality, while achieving an overall EER that is lower than the baseline. A major advantage of our approach is that while both performance discrepancy and overall error is reduced, no explicit information about group membership is required for model training. Adversarial reweighting based on pseudo-labeling, as a function of either the anchor speaker alone or of speaker pairs, showed the lowest overall EER. It is notable that using pairwise weighting can further decrease the standard deviation across different groups. Future work will include a decoupling of the learner and the adversary model, unlike the current model structure in which the adversary network takes its input from the learner. In addition, an analysis on a larger database with balanced nationality groups would be desirable. Our approach could also be extended to other speech characterization tasks, such as for emotion recognition or medical diagnostics.
5. References


