



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 reformulated for speaker verification with metric learning, and shown to improve results across different subgroups of gender and nationality, without requiring annotation of subgroups in the training data. An adversarial network learns a weight for each training sample in the batch so that the main learner is forced to focus on poorly performing instances. Using a min-max optimization algorithm, this method improves overall speaker verification 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 reduced 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, fairness.

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 Amazon 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 assumptions are incorrect, this can create its own biases relative to attributes such as gender or nationality. Also, obtaining the required annotations may be difficult due to privacy concerns [5]. To address the fairness problem, many researches have been focusing on identifying the biases in the training data [5, 6]. These include applications to face recognition [10, 11], image classification [12], speech recognition [13], natural language processing [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 networks trained on subgroups (e.g., male and female speakers) can improve speaker verification fairness, including for unbalanced 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 Laoti *et al.* [21]. Similarly to the distributionally robust optimization (DRO) method proposed by Hashimoto *et al.* [22], this approach aims to automatically identify the underperforming groups using an adversarial network, and adjusts their contribution to the training loss. This was shown to not only reduce the performance disparity for minority groups, but to also improve 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]:

$$J(\theta, \phi) = \min_{\theta} \max_{\phi} \sum_{(\mathbf{x}_i, y_i) \in D \times L} \lambda_{\phi}(\mathbf{x}_i, y_i) l(h_{\theta}(\mathbf{x}_i), y_i), \quad (1)$$

where D and L are data and label sets, respectively. Given a data sample \mathbf{x}_i with its label y_i , the loss is computed as $l(h_{\theta}(\mathbf{x}_i), y_i)$ using the learner network parameters θ . The adversarial weight $\lambda_{\phi}(\mathbf{x}_i, y_i)$ is generated by the adversarial network with parameters ϕ . Given a learner network parameter θ , the adversary parameters ϕ are first updated to maximize the weighted loss $J(\theta, \phi)$, which is expected to increase $\lambda_{\phi}(\mathbf{x}_i, y_i)$ for underperforming groups. Then, $J(\theta, \phi)$ is minimized with respect to θ 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 $l(h_{\theta}(\mathbf{x}_i), y_i)$ is determined by the data instance and its label, the loss for metric learning is defined using 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 pairwise weighting (PW). In APS, the adversarial network computes each speaker's weight using the aggregated similarity to other speakers. In PL, each speaker is labeled in an unsupervised 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 evaluate 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 system and the proposed ARW algorithms for speaker verification.

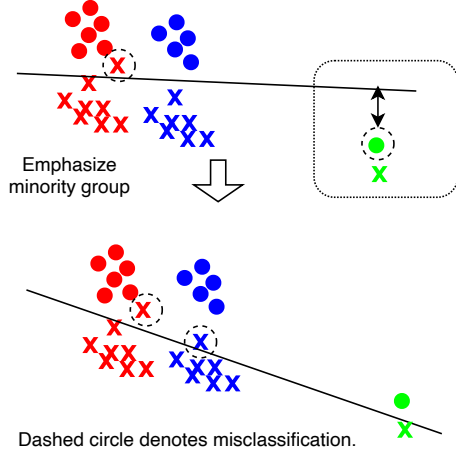


Figure 1: Representation bias in the training data, where each group is colored differently. For binary classification of two different classes of circle and cross, the bottom has better fairness than the top by considering the minority group, green.

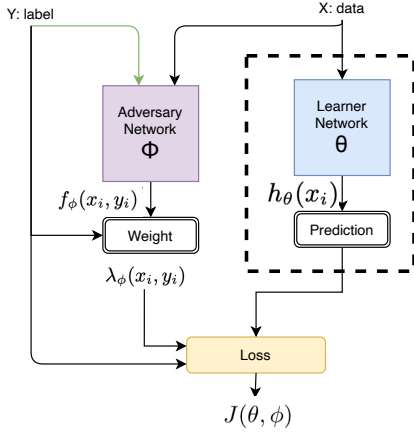


Figure 2: ARW for classification. For inference, only the learner inside the dashed box is used.

Section 3 describes our experiments, and Section 4 summarizes the findings.

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 M utterances from N speakers. For AP loss, we compute the anchor \mathbf{x}_j^a and the query \mathbf{x}_j^q for speaker j :

$$\mathbf{x}_j^a = \frac{1}{M-1} \sum_{m=0}^{M-2} \mathbf{x}_{j,m}, \quad (2)$$

$$\mathbf{x}_j^q = \mathbf{x}_{j,M-1}, \quad (3)$$

where $\mathbf{x}_{j,m}$ for $m \in \{0, \dots, M-1\}$ denotes the embed-

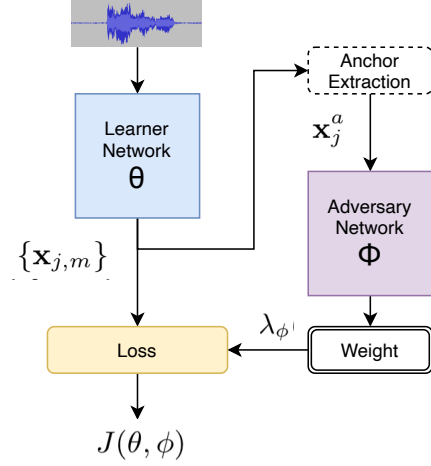


Figure 3: ARW for speaker verification. We define the weight λ_ϕ differently for APS, PL, and PW.

ding extracted from the m th utterance from the speaker $j \in \{0, \dots, N-1\}$. Using the anchor and the query, the similarity between a pair of speakers (j, k) is computed as follows:

$$\mathbf{S}_{j,k} = w \cos(\mathbf{x}_j^a, \mathbf{x}_k^q) + b, \quad (4)$$

where

$$\cos(\mathbf{x}_j^a, \mathbf{x}_k^q) = \frac{\mathbf{x}_j^a \cdot \mathbf{x}_k^q}{\|\mathbf{x}_j^a\| \|\mathbf{x}_k^q\|} \quad (5)$$

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

$$L_p = \frac{1}{N} \sum_{j=0}^{N-1} L_{p,j}, \quad (6)$$

where

$$L_{p,j} = -\log \frac{e^{\mathbf{S}_{j,j}}}{\sum_{k=0}^{N-1} e^{\mathbf{S}_{j,k}}}. \quad (7)$$

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

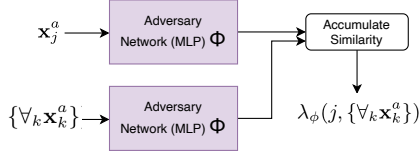


Figure 4: Computing ARW weights $\lambda_\phi(j, \{\forall_k \mathbf{x}_k^a\})$ for APS.

the elements of the ARW algorithm for a classification task. Given input data \mathbf{x}_i and label y_i , the learner makes a prediction of the class y_i . In [21], the weight $f_\phi(\mathbf{x}_i, y_i)$ is computed using a sigmoid for each class of Y so that it ranges over $[0, 1]$, where ϕ is the set of parameters of the adversary network. Finally, the weight $\lambda_\phi(\mathbf{x}_i, y_i)$ is normalized:

$$\lambda_\phi(\mathbf{x}_i, y_i) = 1 + \frac{f_\phi(\mathbf{x}_i, y_i)}{\frac{1}{B} \sum_{j=0}^{B-1} f_\phi(\mathbf{x}_j, y_j)}, \quad (8)$$

where B is the batch size. The training is done as in (1), where the adversary network ϕ tries to maximize the loss (i.e., up-weighting the worst inputs or groups) and the learner (parameterized by θ) 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 ϕ and minimization of $J(\theta, \phi)$ with respect to θ . In this original formulation, the adversary model output $f_\phi(\mathbf{x}_i, y_i)$ 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 in the original ARW, the adversary network f_ϕ 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:

$$J(\theta, \phi) = \min_{\theta} \max_{\phi} \sum_{\mathbf{x}_j, k \in D} \left(\sum_{j=0}^{N-1} \lambda_\phi(j, \{\forall_k \mathbf{x}_k^a\}) L_{p,j} \right). \quad (9)$$

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

$$\lambda_\phi(j, \{\forall_k \mathbf{x}_k^a\}) = 1 + \frac{\sum_k f_\phi(\mathbf{x}_j^a) \cdot f_\phi(\mathbf{x}_k^a)}{\frac{1}{N} \sum_j \sum_k f_\phi(\mathbf{x}_j^a) \cdot f_\phi(\mathbf{x}_k^a)}, \quad (10)$$

and one based on cosine similarity:

$$\lambda_\phi(j, \{\forall_k \mathbf{x}_k^a\}) = 1 + \frac{\sum_k e^{\cos(f_\phi(\mathbf{x}_j^a), f_\phi(\mathbf{x}_k^a))}}{\frac{1}{N} \sum_j \sum_k e^{\cos(f_\phi(\mathbf{x}_j^a), f_\phi(\mathbf{x}_k^a))}}. \quad (11)$$

Here $f_\phi(\mathbf{x}_j^a) \in \mathbb{R}^H$ 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.

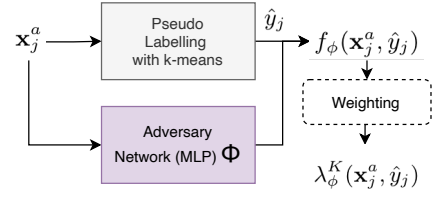


Figure 5: Computing weights $\lambda_\phi^K(\mathbf{x}_j^a)$ for PL.

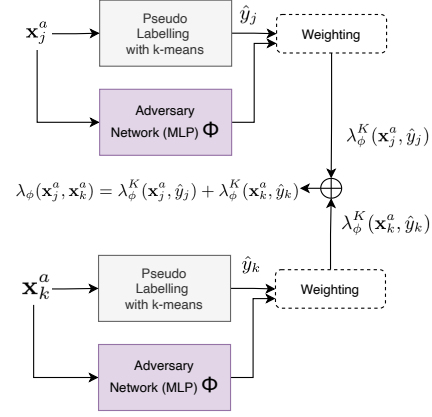


Figure 6: Computing ARW weights $\lambda_\phi(\mathbf{x}_j^a, \mathbf{x}_k^a)$ for PW.

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

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:

$$\hat{y}_j = \arg \max_{k \in \{0, \dots, K-1\}} \cos(\mathbf{c}_k, \mathbf{x}_j^a), \quad (12)$$

where \mathbf{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}_k . 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_ϕ 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, $f_\phi(\mathbf{x}_j^a, \hat{y}_j)$ for $\hat{y}_j \in \{0, \dots, K-1\}$. This leads to

$$J(\theta, \phi) = \min_{\theta} \max_{\phi} \sum_{\mathbf{x}_j, k \in [D]} \left(\sum_{j=0}^{N-1} \lambda_\phi^K(\mathbf{x}_j^a, \hat{y}_j) L_{p,j} \right), \quad (13)$$

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

$$\lambda_\phi^K(\mathbf{x}_j^a, \hat{y}_j) = 1 + \frac{f_\phi(\mathbf{x}_j^a, \hat{y}_j)}{\frac{1}{N} \sum_{k=0}^{N-1} f_\phi(\mathbf{x}_k^a, \hat{y}_k)}. \quad (14)$$

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

2.2.3. Pairwise weighting (PW)

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:

$$J(\theta, \phi) = \min_{\theta} \max_{\phi} \sum_{\mathbf{x}_{j,k} \in [D]} \left(-\frac{1}{N} \sum_{j=0}^{N-1} \log \frac{e^{\lambda_{\phi}(\mathbf{x}_j^a, \mathbf{x}_j^a) \mathbf{S}_{j,j}}}{\sum_{k=1}^N e^{\lambda_{\phi}(\mathbf{x}_j^a, \mathbf{x}_k^a) \mathbf{S}_{j,k}}} \right), \quad (15)$$

and

$$J(\theta, \phi) = \min_{\theta} \max_{\phi} \sum_{\mathbf{x}_{j,k} \in [D]} \left(-\frac{1}{N} \sum_{j=0}^{N-1} \log \frac{\lambda_{\phi}(\mathbf{x}_j^a, \mathbf{x}_j^a) e^{\mathbf{S}_{j,j}}}{\sum_{k=1}^N \lambda_{\phi}(\mathbf{x}_j^a, \mathbf{x}_k^a) e^{\mathbf{S}_{j,k}}} \right). \quad (16)$$

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 $\lambda_{\phi}(\mathbf{x}_j^a, \mathbf{x}_k^a)$ as a combination of the single-speaker PL weights (14) to avoid doubling the input dimensionality:

$$\lambda_{\phi}(\mathbf{x}_j^a, \mathbf{x}_k^a) = \lambda_{\phi}^K(\mathbf{x}_j^a, \hat{y}_j) + \lambda_{\phi}^K(\mathbf{x}_k^a, \hat{y}_k), \quad (17)$$

where \hat{y}_j and \hat{y}_k 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 adversary 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

Table 1: Overall EERs (%) and gender-group EER (%) for APS, PL, and PW.

Method	ALL	female (45%)	male (55%)	gap
Baseline	1.17	0.69	1.39	0.70
APS (10)	1.09	0.67	1.29	0.62
APS (11)	1.12	0.65	1.26	0.61
PL	1.08	0.63	1.27	0.64
PW (15)	1.09	0.65	1.27	0.62
PW(16)	1.08	0.67	1.25	0.58

Table 2: EER by nationality (%).

Method	US (64%)	UK (17%)	Others (19%)	std.
Baseline	1.09	0.72	1.22	0.21
APS (10)	1.05	0.72	1.24	0.21
APS (11)	1.12	0.80	1.26	0.19
PL	1.07	0.69	1.16	0.20
PW (15)	1.09	0.80	1.26	0.19
PW (16)	1.04	0.76	1.22	0.19

(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 lumps 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-labelling, 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

- [1] D. A. Reynolds, "An overview of automatic speaker recognition technology," in *Proc. IEEE ICASSP*, vol. 4, 2002, pp. 4072–4075.
- [2] J. H. L. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.
- [3] A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel, "Racial disparities in automated speech recognition," *Proceedings of the National Academy of Sciences*, vol. 117, no. 14, pp. 7684–7689, 2020.
- [4] Y. Wu, D. Rough, A. Bleakley, J. Edwards, O. Cooney, P. R. Doyle, L. Clark, and B. R. Cowan, "See what I'm saying? Comparing intelligent personal assistant use for native and non-native language speakers," in *Proc. 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2020, pp. 1–9.
- [5] S. Caton and C. Haas, "Fairness in machine learning: A survey," arXiv preprint arXiv:2010.04053, Oct. 2020.
- [6] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.
- [7] F. Kamiran and T. Calders, "Classification with no discrimination by preferential sampling," in *Proc. 19th Machine Learning Conference of Belgium and The Netherlands*, 2010, pp. 1–6.
- [8] S. Bej, N. Davtyan, M. Wolfien, M. Nassar, and O. Wolkenhauer, "LoRAS: An oversampling approach for imbalanced datasets," *Machine Learning*, vol. 110, no. 2, pp. 279–301, 2021.
- [9] M. Gupta, A. Cotter, M. M. Fard, and S. Wang, "Proxy fairness," arXiv preprint arXiv:1806.11212, Jun. 2018.
- [10] S. Samadi, U. Tantipongpipat, J. H. Morgenstern, M. Singh, and S. Vempala, "The price of fair PCA: One extra dimension," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [11] P. Terhörst, J. N. Kolf, M. Huber, F. Kirchbuchner, N. Damer, A. Morales, J. Fierrez, and A. Kuijper, "A comprehensive study on face recognition biases beyond demographics," *IEEE Transactions on Technology and Society*, vol. 3, no. 1, pp. 16–30, Sep. 2021.
- [12] S. Hwang, S. Park, P. Lee, S. Jeon, D. Kim, and H. Byun, "Exploiting transferable knowledge for fairness-aware image classification," in *Proc. Asian Conference on Computer Vision*, 2020.
- [13] Z. Liu, I.-E. Veliche, and F. Peng, "Model-based approach for measuring the fairness in ASR," in *Proc. IEEE ICASSP*, 2022, pp. 6532–6536.
- [14] N. Swinger, M. De-Arteaga, N. T. Heffernan IV, M. D. Leiserson, and A. T. Kalai, "What are the biases in my word embedding?" in *Proc. AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 305–311.
- [15] H. M. Thompson, B. Sharma, S. Bhalla, R. Boley, C. McCluskey, D. Dligach, M. M. Churpek, N. S. Karnik, and M. Afshar, "Bias and fairness assessment of a natural language processing opioid misuse classifier: detection and mitigation of electronic health record data disadvantages across racial subgroups," *Journal of the American Medical Informatics Association*, vol. 28, no. 11, pp. 2393–2403, 2021.
- [16] A. K. Menon and R. C. Williamson, "The cost of fairness in binary classification," in *Proc. Conference on Fairness, Accountability and Transparency*. PMLR, 2018, pp. 107–118.
- [17] L. Oneto, M. Doninini, A. Elders, and M. Pontil, "Taking advantage of multitask learning for fair classification," in *Proc. AAAI/ACM Conference on AI, Ethics, and Society*, 2019, pp. 227–237.
- [18] G. Fenu, M. Marras, G. Medda, and G. Meloni, "Fair voice biometrics: Impact of demographic imbalance on group fairness in speaker recognition," in *Proc. Interspeech*, 2021, pp. 1892–1896.
- [19] W. Toussaint and A. Y. Ding, "SVEva Fair: A framework for evaluating fairness in speaker verification," arXiv preprint arXiv:2107.12049, Jul. 2021.
- [20] H. Shen, Y. Yang, G. Sun, R. Langman, E. Han, J. Droppo, and A. Stolcke, "Improving fairness in speaker verification via group-adapted fusion network," in *Proc. IEEE ICASSP*, 2022.
- [21] P. Lahoti, A. Beutel, J. Chen, K. Lee, F. Prost, N. Thain, X. Wang, and E. Chi, "Fairness without demographics through adversarially reweighted learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 728–740, 2020.
- [22] T. Hashimoto, M. Srivastava, H. Namkoong, and P. Liang, "Fairness without demographics in repeated loss minimization," in *Proc. International Conference on Machine Learning*. PMLR, 2018, pp. 1929–1938.
- [23] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proc. 31st International Conference on Neural Information Processing Systems*, Dec. 2017, pp. 4080–4090.
- [24] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Proc. Interspeech*, Sep. 2018, pp. 1086–1090.
- [25] J. S. Chung, J. Huh, and S. Mun, "Delving into VoxCeleb: Environment invariant speaker recognition," in *Proc. Odyssey Speaker and Language Recognition Workshop*, 2020, pp. 349–356.
- [26] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Proc. Interspeech*, 2020, pp. 2977–2981.
- [27] H. S. Heo, B.-J. Lee, J. Huh, and J. S. Chung, "Clova baseline system for the VoxCeleb speaker recognition challenge 2020," arXiv preprint arXiv:2009.14153, Sep. 2020.
- [28] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, Oakland, CA, 1967, pp. 281–297.
- [29] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, Aug. 2017, pp. 2616–2620.
- [30] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *Proc. Interspeech 2018*, pp. 2252–2256, 2018.