



Extrapolating False Alarm Rates in Automatic Speaker Verification

Alexey Sholokhov¹, Tomi Kinnunen², Ville Vestman², Kong Aik Lee³

¹Huawei Technologies Ltd., Moscow, Russia

²Computational Speech Group, University of Eastern Finland, Finland

³Biometrics Research Laboratories, NEC Corporation, Tokyo, Japan

sholokhov.alexey@huawei.com, tkinnu@cs.uef.fi, vvestman@cs.uef.fi, kongaik.lee@nec.com

Abstract

Automatic speaker verification (ASV) vendors and corpus providers would both benefit from tools to reliably extrapolate performance metrics for large speaker populations *without collecting new speakers*. We address false alarm rate extrapolation under a worst-case model whereby an adversary identifies the closest impostor for a given target speaker from a large population. Our models are generative and allow sampling new speakers. The models are formulated in the ASV detection score space to facilitate analysis of arbitrary ASV systems.

Index Terms: speaker verification, false alarm rate, closest impostor, black-box attack, PLDA, implicit generative models

1. Introduction

How unique the human voice is? This question is clearly relevant for practical deployment of automatic speaker verification (ASV) technology — yet, scarcely addressed [1] due to the open-ended nature of the question. Unlike passwords that have zero uncertainty conditioned on person’s identity [2], the human voice is subject to both extrinsic and intrinsic variations, none of which are deterministic. ‘Uniqueness’ thus depends both on data conditions *and* the observer (*e.g.* a specific ASV system or listener). In our recent work [3], we addressed an alternative, more tangible question:

Given a specific ASV system (black-box) and evaluation corpus, how does the false alarm rate behave with an increased number of speakers?

To be precise, we modeled the sampling process of nontarget detection scores of a given ASV system through a probabilistic generative model to enable indefinite increasing of the impostor population size without having to collect new speech data. The assumption is that the underlying sampling process, governed by the properties of the ASV system (treated as a black-box) and corpus, remains fixed. Drawing a random nontarget score proceeds in two steps. First, we draw a *random pair of speakers* implicitly represented by a Gaussian distribution which models similarity scores between these two speakers. Second, we draw a *random score* from that distribution.

In [3] we also revised the notion of ‘nontarget speaker’. Apart from efforts devoted to the study of spoofing attacks [4], standard evaluation benchmarks of ASV technology [5] assume nontarget speakers to be *non-proactive* or *zero-effort* impostors — other *random* speakers paired up with targets. We, instead, considered *worst-case* impostors with a *deterministic*, proactive imposture policy: given a target speaker of interest (for instance, a notable politician), the adversary identifies the closest impostor to the given target from a large population (such as the Internet) to increase the chance of this impostor to be accepted as the targeted speaker. This is an instance of *adversarial attack* [6, 7] on ASV [8, 9]. The general motivations are to identify

loopholes of ASV and to develop defense mechanisms against them.

In this study we improve upon the generative model presented in [3]. Despite demonstrating expected overall trends, the predicted false alarm rates were substantially overestimated, particularly at high ASV thresholds (proxies of high-security applications). To tackle this shortcoming, we propose a discriminative training method which uses empirical estimates of false alarm rates as targets. The setup is similar to standard regression tasks except that our primary goal is *extrapolation* — making predictions substantially beyond the range of inputs in the training set. In our context, this means predicting false alarm rate of an ASV system for a population of, say, 100,000 nontarget speakers but with access to data from only 1000 speakers. Without additional assumptions on the predictor functions, standard regression methods available in machine learning libraries have higher risk of producing meaningless results (see [10]).

In general, the task of learning *interpretable* functional dependencies has received far less attention within machine learning compared to natural sciences, where discovering *physically plausible* models is important. To obtain more trustworthy predictions, we build upon a regressor which takes into account the specifics of detection score distribution governed by unobserved similarities between speakers. Specifically, it uses a generative model of ASV scores together with an estimator of false alarm rates in a single prediction pipeline.

Another novelty of this work is modeling the generation of nontarget scores using *probabilistic linear discriminant analysis* (PLDA) **in detection score space**. PLDA [11] — a generative model in the space of vector representations of speech utterances (*e.g.* i-vectors or x-vectors) — is well-known by ASV researchers. Our formulation, however, differs substantially from this familiar use case as our modeling takes place in the detection score, rather than vector space. We use PLDA to generate ‘new’ detection scores. The scores used for training can, but are not required to be outcomes of trial comparisons by an actual PLDA model. We learn PLDA whose log-likelihood ratio scores are designed to approximate distribution of detection scores of *any* ASV system. Similar to [3], our models require no other data than ASV scores (and their labels). In specific, we do *not* need any speaker embeddings to train our PLDA score generator.

2. Preliminaries

We begin with a brief review of some necessary technical background on false alarm rate, its extrapolation, and PLDA.

2.1. False alarm rate

The *false alarm* (FA) rate is defined as

$$P_{\text{FA}}(\tau) \equiv \int_{\tau}^{\infty} p(s|\text{non}) ds, \quad (1)$$

where $\tau \in \mathbb{R}$ is a detection threshold and $p(s|\text{non})$ is the probability density of nontarget scores of an ASV system. The FA rate can be written as the expectation $\mathbb{E}_{s \sim p(s|\theta_{\text{non}})}[\mathbb{I}\{s > \tau\}]$, and approximated by Monte-Carlo (MC) sampling as

$$P_{\text{FA}}(\tau) \approx \frac{1}{R} \sum_{r=1}^R \mathbb{I}\{s_r > \tau\}, \quad s_r \sim p(s|\theta_{\text{non}}), \quad (2)$$

where $r = 1, \dots, R$ are the indices of the nontarget trials and $\mathbb{I}\{\cdot\}$ is an indicator function. Each nontarget trial consists of a pairwise comparison of utterances from two different speakers (conversely, a target trial constitutes a pairwise comparison of utterances from the same speaker). In the special case when every unique speaker pair in a trial list has the same number of trials, L , the above estimator is the same as averaging *speaker-pair specific* FA rates:

$$\frac{1}{R} \sum_{r=1}^R \mathbb{I}\{s_r > \tau\} = \frac{1}{T} \sum_{i=1}^T \frac{1}{L} \sum_{\ell=1}^L \mathbb{I}\{s_{i,\ell} > \tau\} \quad (3)$$

where $s_{i,\ell}$ denotes the ℓ^{th} score from the i^{th} speaker pair, T is the number of unique speaker pairs and $R = T \cdot L$.

This reformulation of (2) leads to an alternative estimator of P_{FA} as presented in [3]:

$$P_{\text{FA}}(\tau) \approx \frac{1}{T} \sum_{i=1}^T P_{\text{FA}}^{(i)}(\tau), \quad P_{\text{FA}}^{(i)}(\tau) = \frac{1}{|\mathcal{S}_i|} \sum_{s_\ell \in \mathcal{S}_i} \mathbb{I}\{s_\ell > \tau\}, \quad (4)$$

where \mathcal{S}_i is the set of scores for the i^{th} speaker pair consisting of an enrolled (target) speaker and an impostor selected randomly from a dataset and $P_{\text{FA}}^{(i)}(\tau)$ is the corresponding speaker-pair specific FA rate. The following discussion is based on the fact that selecting a random impostor is equivalent to selecting a random subset of N speakers, followed by selecting a random speaker from this subset. Thus, (4) can be interpreted as averaging of results of T stochastic simulations, where both the target speaker and the impostor subset are randomly drawn from a given database. This is in line with typical ASV trial designs, where (zero-effort) impostors can be considered as random speakers with different identity.

This view of (4) allows us to consider several alternative policies to choose an impostor. We consider *worst-case* impostor that is the closest match to a given target speaker. The adversary might locate the closest impostor using a speaker identification system [12] or by other means. In [3] we proposed a new metric, *worst-case FA rate with N impostors*, abbreviated $P_{\text{FA}}^N(\tau)$. It represents a scenario where target speakers are scored against their closest impostors. We also introduced generative model of nontarget scores to allow N to exceed the number of speakers in the corpus. This allows extrapolation of FA rates for arbitrarily-sized impostor population.

2.2. Probabilistic linear discriminant analysis (PLDA)

PLDA [11] models between- and within-class distributions of high-dimensional vectors using low-dimensional subspaces. In ASV, PLDA is used to model distributions of *speaker embeddings* and for same/different speaker hypothesis testing. PLDA was revised in [13] and [14] (see also [15]). We use the so-called *two-covariance* PLDA [14]. It models the j th embedding of the i th speaker by

$$\phi_{i,j} = \mathbf{b} + \mathbf{y}_i + \boldsymbol{\varepsilon}_{i,j}, \quad (5)$$

where $\mathbf{b} \in \mathbb{R}^D$ is the center of the embedding space, $\mathbf{y}_i \in \mathbb{R}^D$ is a latent *speaker identity variable* with normal prior $\mathcal{N}(\mathbf{0}, \mathbf{B})$, and $\boldsymbol{\varepsilon}_{i,j} \in \mathbb{R}^D$ is *residual* with prior $\mathcal{N}(\mathbf{0}, \mathbf{W})$. \mathbf{B} and \mathbf{W} are the *between-* and *within-*class covariance matrices. The pa-

rameters $\boldsymbol{\theta}_{\text{plda}} = \{\mathbf{b}, \mathbf{B}, \mathbf{W}\}$ are typically estimated via the *expectation-maximization* (EM) algorithm [16, 17] using a set of development speakers (different from target speakers).

At the recognition stage, $\boldsymbol{\theta}_{\text{plda}}$ is used for computing *log-likelihood ratio* (LLR) score for a given pair of enrollment and test utterances, as

$$s(\phi_e, \phi_t) = \log \frac{p(\phi_e, \phi_t | H_0, \boldsymbol{\theta}_{\text{plda}})}{p(\phi_e, \phi_t | H_1, \boldsymbol{\theta}_{\text{plda}})}, \quad (6)$$

where H_0 and H_1 denote, respectively, the target (same speaker) and nontarget (different speaker) hypotheses. H_0 assumes that ϕ_e and ϕ_t share the same latent identity variable and H_1 assumes that their latent identity variables are different. The score (6) is given by a closed-form expression — see [18].

2.3. PLDA in the score space

In this work, we do *not* use PLDA to model speaker embeddings. For generality, all our modeling takes place in the detection scores space. We use PLDA to model the distribution of empirical scores of *any* ASV system — whether or not based on a PLDA back-end. Note, first, that (6) represents a deterministic function $s: \mathbb{R}^{2D} \rightarrow \mathbb{R}$ that assigns a real number to any pair of embeddings. Concerning performance assessment, the embeddings are not relevant. The distribution of the detection scores (rather, the *order* of the scores) is a complete description of the detection error trade-off (DET) behavior of a given system [19]. Second, note that PLDA is a generative model — it allows sampling new ‘speakers’ in the \mathbf{y} -space. We want to fit a PLDA model whose score generation mechanism produces distributions similar to the given empirical scores.

To this end, we first note that PLDA is heavily overparameterized from the perspective of LLR score order preservation. A centered PLDA model ($\mathbf{b} = \mathbf{0}$) uses $D(D+1)/2$ parameters for each of the matrices \mathbf{B} and \mathbf{W} , totaling $D^2 + D$ [15]. In fact, we need only D numbers. Note that any invertible linear transformation of the feature space leaves the order of scores unchanged. Hence, it does not alter a DET-curve. We can therefore perform *simultaneous diagonalization* [20, 21] of the within-class and between-class covariance matrices such that (i) \mathbf{B} becomes an identity matrix and (ii) \mathbf{W} becomes diagonal: $\mathbf{W} = \text{diag}(d_1, \dots, d_D)$. Therefore, a PLDA model can be defined through D nonnegative numbers. We use this minimal parametrization in our experiments.

3. Extrapolating false alarm rates

With the above preliminaries, we are now set to present models to produce predictions of $P_{\text{FA}}^N(\tau)$. We consider two different types of models. Our previous model [3] is a special case of a **location-scale** model described below, while **PLDA-based** model is a new proposal. Both models serve to approximate the distribution of *sets of scores* between a random target speaker and the closest impostor selected from a random set of N impostors. These sets of scores can be viewed as outcomes of the generative process in Algorithm 1.

Here, $\text{sim}(\cdot, \cdot)$ is any speaker similarity measure. Since explicit speaker representation are not available in the general case the similarity function has to be computed from a set of speaker-pair specific scores. This case includes estimating $P_{\text{FA}}^N(\tau)$ from empirical scores. We use the mean value of scores as a similarity measure. Given a sampled *set* of score sets $\{\mathcal{S}_1, \dots, \mathcal{S}_T\}$, we compute the corresponding MC estimates of the speaker-pair conditioned FA rates $\{P_{\text{FA}}^{(1)}(\tau), \dots, P_{\text{FA}}^{(T)}(\tau)\}$ — the individual terms of the sum in (4). Averaging them yields an estimate of $P_{\text{FA}}^N(\tau)$. We now describe two generative models that

Algorithm 1

for $i = 1, \dots, T$ **do**

Sample random enrolled (target) speaker, $\mathbf{y}_e^{(i)}$.
Sample N random test speakers, $\mathbf{y}_{t,1}^{(i)}, \mathbf{y}_{t,2}^{(i)}, \dots, \mathbf{y}_{t,N}^{(i)}$.
Find the closest speaker $\mathbf{y}_{t,k}^{(i)}$, where
 $k = \arg \max_j \text{sim}(\mathbf{y}_e^{(i)}, \mathbf{y}_{t,j}^{(i)})$
Sample scores $\mathcal{S}_i = \{s_\ell\}_{\ell=1}^L$ between $\mathbf{y}_e^{(i)}$ and $\mathbf{y}_{t,k}^{(i)}$.

end

allow sampling scores according to Algorithm 1 for an arbitrary N . Each model can be trained on sets of speaker-pair specific ASV scores and further be used for FA rate extrapolation.

3.1. Location-scale models

Our first family of models assumes the distribution of between-speaker scores for a given pair of speakers to be a scaled and shifted version of some base distribution defined by its *cumulative distribution function* (CDF). Our earlier model [3] assumes a Gaussian base distribution. The following generalized algorithm allows to generate a set of between-speaker scores for given N :

1. Sample N pairs of location-scale values $\{(\mu_j, \sigma_j)\}_{j=1}^N$
2. Find the largest location parameter $\mu_k = \max_j \{\mu_j\}$
3. Sample scores $\mathcal{S}_i = \{s_\ell\}_{\ell=1}^L$ by $s_\ell = \mu_k + \sigma_k F^{-1}(u_\ell)$, where $u_\ell \sim U[0, 1]$ is uniformly-distributed.

Here, $F(\cdot)$ is the CDF of the base distribution of scores. The algorithm uses *inverse transform sampling* [22] to generate scores from the underlying distribution. Each pair (μ_j, σ_j) parameterizes the distribution of scores between a fixed target speaker and the j th impostor. It also assumes that the closest impostor has the largest location parameter μ_j . One limitation of the model in [3] is the unrealistic assumption of Gaussian between-speaker scores. Here $F(\cdot)$ is allowed to be arbitrary. In practice, we use `torchpwl`¹ to define a piece-wise linear function with monotonicity constraint for CDF approximation.

3.2. PLDA-based model

The above location-scale family of models represent speakers indirectly through their *relative similarities* defined through between-speaker score distributions. The model described in this section uses, instead, latent identity variables to represent individual speakers *explicitly*. This gives an alternative predictor of $P_{\text{FA}}^N(\tau)$ based on PLDA.

A PLDA model with known parameters θ_{plda} can be used to generate LLR scores, as follows.

1. Sample a pair of enrollment and test latent identity variables $(\mathbf{y}_e, \mathbf{y}_t)$ from the prior: $\mathbf{y}_e \sim \mathcal{N}(\mathbf{0}, \mathbf{B})$ and $\mathbf{y}_t = \mathbf{y}_e$ under H_0 ; or draw the second sample $\mathbf{y}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{B})$ under H_1 .
2. Sample a pair of enrollment and test feature vectors $\phi_e \sim \mathcal{N}(\mathbf{y}_e, \mathbf{W})$, $\phi_t \sim \mathcal{N}(\mathbf{y}_t, \mathbf{W})$ conditioned on the latent identity variables from the first step.
3. Compute the LLR score as $s = \ell(\phi_e, \phi_t)$ using (6).

Note that the two first steps are stochastic, while the LLR score is a deterministic function of the sampled pair of feature vectors and the PLDA model. Under the H_1 hypothesis,

¹<https://pympi.org/project/torchpwl/>

this generative procedure yields scores of zero-effort impostors. Following Algorithm 1, it can be extended to sample $N > 1$ impostors for a given target speaker. To select the closest impostor, we use the LLR score (6) as a similarity measure between speakers. Given identity variable of the closest impostor, one can sample a set of speaker-pair specific scores by repeating steps 2 and 3 in the algorithm above.

We include a learnable monotonic warping function applied to the scores generated by the model to increase flexibility.

3.3. Training methods

We now describe a method for training generative models introduced above. The training data is a set of sets of between-speaker scores produced by any ASV system (black-box). Generally, there are at least three alternative approaches to construct a regressor for predicting $P_{\text{FA}}^N(\tau)$, given N and τ . The first one is to use any standard general-purpose regression technique to match model predictions and empirical estimates of $P_{\text{FA}}^N(\tau)$ computed with (4) from the empirical scores. Despite the apparent simplicity and attractiveness of such approach, and due to the lack of task-specific constraints, such models are exposed to a greater risk of failure for large values of N [10].

The second approach is to follow a two-stage strategy: First, train a generative model of scores and use it to generate nontarget scores following Algorithm 1. Next, use the generated sets of scores to estimate $P_{\text{FA}}^N(\tau)$ using (4). We used this approach in [3] where a location-scale model with Gaussian base distribution was trained to maximize model log-likelihood [17]. Different from [3], the models proposed in this work are instances of *implicit* generative models: they are specified through a forward stochastic procedure for data generation, but do not allow direct likelihood evaluation [23, 24]. Even if implicit generative models can be trained using plethora of methods different from ML estimation (see [23] and [25]), it is non-trivial to design a training algorithm for models whose training set is a *set of sets* (see, e.g. [26]), as is the case here.

In the last approach, generative model is also included to the prediction pipeline but trained *discriminatively* by comparing the model-based estimates of $P_{\text{FA}}^N(\tau)$ against the corresponding empirical estimates (treated as ground-truth). The regressor is trained by minimizing the mean square error (MSE) between the empirical and model-based false alarm rates. To address lack of differentiability, we replace the unit step function in (4) by the sigmoid function with scaled argument. Also, the `argmax` function which appears in PLDA score generating algorithm was replaced by its approximation computed as a weighted sum of the speaker identity variables where weights are softmax-normalized similarities to the target speaker. This is similar to a so-called soft-attention mechanism introduced in [27].

In contrast to purely generative training aimed at approximating distribution of scores, discriminative training optimizes directly the final regression target. Using a restricted class of regression functions, in turn, allows us to keep the extrapolated values within the range of reasonable expectations.

The resulting objective function (MSE in our experiments) includes random sampling and can be viewed as a *nested* Monte-Carlo estimate of the expected loss. Generally, such MC estimates are biased for any finite T [28] but useful for training via stochastic optimization, provided that T is sufficiently large (100-1000 in our experiments). We used Adam [29] optimizer with mini-batches of size 20 and learning rate 10^{-3} to train both models.

4. Experiments

We closely follow the experimental setup of [3]. We combine Voxceleb1 [30] and Voxceleb2 [31] corpora to have a dataset with a large number of speakers and sufficient number of utterances per speaker needed for reliable estimation of P_{FA}^N . The resulting dataset has 7365 speakers with more than 100 utterances from each speaker, on average. The data was divided into three disjoint sets with 5345, 40 and 2000 speakers. The first set was used to train the ASV systems. The second one is the standard Voxceleb1 evaluation protocol [32], used as a sanity check of our ASV systems (see [3] for details). The third set which contains 1000 male and 1000 female speakers was used to compute scores for training models for P_{FA}^N extrapolation. We computed similarity scores for each unique speaker pair of the same gender. To this end, we randomly selected 18 utterances for each of 2000 speakers to obtain at least three hundreds of scores ($18^2 = 324$), which we assume to be sufficient to represent speaker-pair specific score distributions.

We used two standard ASV systems based on i-vectors and x-vectors to compute ASV scores used in our experiments. Due to the space limitations, we present results only for the x-vector system, which has EER of 3.61% on the standard Voxceleb1 evaluation protocol. The key conclusions, however, are similar for the i-vector system. For more details on ASV systems and setup, refer to [3].

We computed empirical and model-based estimates of the worst-case false alarm rates with N impostors, P_{FA}^N , by randomly selecting a target speaker $T = 1000$ times in Algorithm 1. Fig. 1 shows the estimates obtained with different models. The three groups of curves correspond to different choices of ASV threshold, τ . As detailed in [3], these thresholds are the minimizers of three different detection cost functions (DCF). The first DCF has high cost for misses (τ_1), the second DCF has equal costs for misses and false alarms (τ_2), and the last one penalizes false alarms more (τ_3). The empirical curves end up to $N = 1000$ impostors (as we have exhausted all data) while the extrapolated regression curves for $N > 1000$ may be used to speculate about the range of values of P_{FA}^N for large sizes of impostor population. For instance, the ASV system with $P_{FA}^1 = P_{FA} \approx 1\%$ may have the worst case false alarm rate around 50% for $N = 10^5$. That is, if the attacker has a speech sample of the target speaker *and* access to a proxy ASV system with comparable accuracy to the attacked one, the chance of accepting the closest impostor may reach 50% for a population of 10^5 available impostors.

To objectively assess the quality of models' forecasts, we measure mean absolute error (MAE) on the extrapolated values of P_{FA}^N for a held-out set with $N \in [660, 999]$ while the corresponding empirical values (treated as the ground-truth and computed according to (4)) were unseen by the model during training. In specific, the inputs in the training data were formed as pairs (N, τ) uniformly sampled from $[1, 660] \times [\tau_{\min}, \tau_{\max}]$, where the range of thresholds is determined according to the range of empirical scores. The held-out set was created similarly but with a different range of N . The results summarised in Table 1 indicate that more flexible models produce more accurate predictions. For instance, using a learnable base distribution instead of Gaussian decreases MAE for location-scale models and both models benefit from score warping. The location-scale and PLDA models have comparable accuracy. Importantly, both provide substantial improvement over earlier, purely generative model [3].

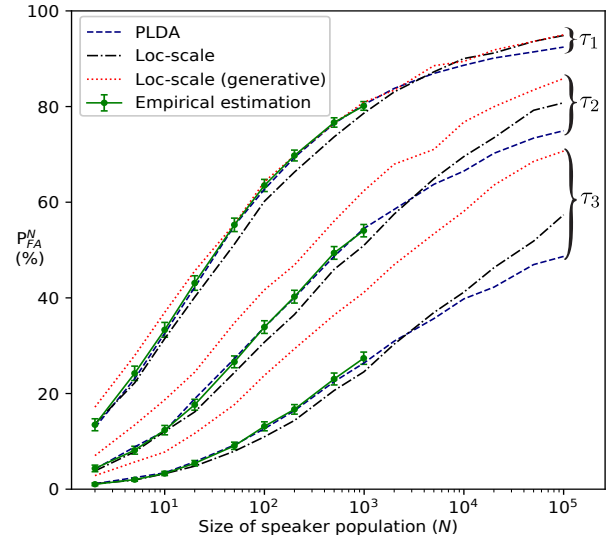


Figure 1: Worst-case false alarm estimates for male scores given by x-vector system. Two new models and the model from [3] are shown along with the empirical estimates. The estimates are shown together with their 99% confidence intervals.

Table 1: Extrapolation performance for different models in terms of MAE computed on the held-out set. + indicates that a learnable score warping function was included to a model. We found that PLDA with 10-dimensional feature space produces the best results.

Model	MAE, %
Location-scale (Gaussian), generative [3]	8.34
Location-scale (Gaussian)	1.34
Location-scale (Gaussian, +)	0.57
Location-scale (general CDF)	0.67
Location-scale (general CDF, +)	0.48
PLDA ($D = 10$)	1.18
PLDA ($D = 10$, +)	0.39

5. Conclusions

We advanced our recent work [12, 3] on worst-case impostors in the context of ASV. In specific, we introduced new tools for performance extrapolation of ASV systems. The models operate on detection score space and are therefore applicable outside the scope of ASV too. Our results indicate substantial improvement over our previous model [3].

In future work, we may relax our worst-case impostor assumption, for instance so that the attacker fails to identify the closest impostor. More generally, the usual assumption in adversarial machine learning where the attacker knows everything of the attacked system is potentially overly-pessimistic.

6. Acknowledgements

This work was supported in part by the Academy of Finland (Proj. No. 309629).

7. References

- [1] A. Nautsch, C. Rathgeb, R. Saeidi, and C. Busch, "Entropy analysis of i-vector feature spaces in duration-sensitive speaker recognition," in *2015 IEEE International Conference on Acoustics,*

- Speech and Signal Processing (ICASSP)*, April 2015, pp. 4674–4678.
- [2] K. Takahashi and T. Murakami, “A measure of information gained through biometric systems,” *Image and Vision Computing*, vol. 32, no. 12, pp. 1194–1203, 2014.
 - [3] A. Sholokhov, T. Kinnunen, V. Vestman, and K. A. Lee, “Voice biometrics security: Extrapolating false alarm rate via hierarchical Bayesian modeling of speaker verification scores,” *Computer Speech & Language*, vol. 60, p. 101024, 2020.
 - [4] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilçi, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado, “ASVspoof: The automatic speaker verification spoofing and countermeasures challenge,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, June 2017.
 - [5] C. S. Greenberg, L. P. Mason, S. O. Sadjadi, and D. A. Reynolds, “Two decades of speaker recognition evaluation at the national institute of standards and technology,” *Computer Speech & Language*, vol. 60, p. 101032, 2020.
 - [6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” in *International Conference on Learning Representations*, 2014. [Online]. Available: <http://arxiv.org/abs/1312.6199>
 - [7] B. Biggio and F. Roli, “Wild patterns: Ten years after the rise of adversarial machine learning,” *Pattern Recognition*, vol. 84, pp. 317–331, 2018.
 - [8] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, “Fooling end-to-end speaker verification with adversarial examples,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1962–1966.
 - [9] R. K. Das, X. Tian, T. Kinnunen, and H. Li, “The attacker’s perspective on automatic speaker verification: An overview,” 2020.
 - [10] G. S. Martius and C. Lampert, “Extrapolation and learning equations,” in *5th International Conference on Learning Representations, ICLR 2017-Workshop Track Proceedings*, 2017.
 - [11] S. Prince, P. Li, Y. Fu, U. Mohammed, and J. H. Elder, “Probabilistic models for inference about identity,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 144–157, 2012.
 - [12] V. Vestman, T. Kinnunen, R. G. Hautamäki, and M. Sahidullah, “Voice mimicry attacks assisted by automatic speaker verification,” *Computer Speech Language*, vol. 59, pp. 36–54, 2020.
 - [13] P. Kenny, “Bayesian speaker verification with heavy-tailed priors,” in *Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic, June 28 – July 1, 2010*, 2010.
 - [14] N. Brümmer and E. de Villiers, “The speaker partitioning problem,” in *Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic, June 2010*, p. 34.
 - [15] A. Sizov, K. Lee, and T. Kinnunen, “Unifying probabilistic linear discriminant analysis variants in biometric authentication,” in *Structural, Syntactic, and Statistical Pattern Recognition - Joint IAPR International Workshop, S+SSPR 2014, Joensuu, Finland, August 20-22, 2014. Proceedings*, 2014, pp. 464–475.
 - [16] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.
 - [17] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.
 - [18] J. Rohdin, S. Biswas, and K. Shinoda, “Discriminative PLDA training with application-specific loss functions for speaker verification,” in *Odyssey 2014: The Speaker and Language Recognition Workshop, Joensuu, Finland, June 16-19, 2014*, 2014, pp. 26–32.
 - [19] N. Brümmer, “Measuring, refining and calibrating speaker and language information extracted from speech,” Ph.D. dissertation, Stellenbosch University, 2010.
 - [20] K. Fukunaga, *Introduction to Statistical Pattern Recognition (2nd Ed.)*. USA: Academic Press Professional, Inc., 1990.
 - [21] Y. Wang, H. Xu, and Z. Ou, “Joint bayesian gaussian discriminant analysis for speaker verification,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5390–5394.
 - [22] L. Devroye, *Non-Uniform Random Variate Generation*. New York, NY, USA: Springer-Verlag, 1986.
 - [23] S. Mohamed and B. Lakshminarayanan, “Learning in implicit generative models,” *CoRR*, vol. abs/1610.03483, 2016. [Online]. Available: <http://arxiv.org/abs/1610.03483>
 - [24] I. J. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks,” *CoRR*, vol. abs/1701.00160, 2017. [Online]. Available: <http://arxiv.org/abs/1701.00160>
 - [25] G. Louppe, J. Hermans, and K. Cranmer, “Adversarial variational optimization of non-differentiable simulators,” in *Proceedings of Machine Learning Research*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and M. Sugiyama, Eds., vol. 89. PMLR, 16–18 Apr 2019, pp. 1438–1447.
 - [26] C.-L. Li, M. Zaheer, Y. Zhang, B. Póczos, and R. Salakhutdinov, “Point cloud GAN,” *arXiv preprint arXiv:1810.05795*, 2018.
 - [27] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
 - [28] T. Rainforth, R. Cornish, H. Yang, A. Warrington, and F. Wood, “On nesting Monte Carlo estimators,” in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 4267–4276.
 - [29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
 - [30] A. Nagrani, J. S. Chung, and A. Zisserman, “VoxCeleb: A large-scale speaker identification dataset,” *Proc. Interspeech 2017*, pp. 2616–2620, 2017.
 - [31] J. S. Chung, A. Nagrani, and A. Zisserman, “VoxCeleb2: Deep speaker recognition,” in *Interspeech*, 2018.
 - [32] J. S. Chung, A. Nagrani, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, “VoxSRC 2019: The first VoxCeleb speaker recognition challenge,” *ISCA Challenges*, 2019.