Clustering-Based Score Normalization for Speaker Verification

Bin Gu*, Wu Guo*, Yao Liu† and Jian Sun*
* University of Science and Technology of China
National Engineering Laboratory for Speech and Language Information Processing, Hefei, China
E-mail: bin2801@mail.ustc.edu.cn, guowu@ustc.edu.cn, sjian17@mail.ustc.edu.cn
† China General Technology Research Institute, Beijing, China
E-mail: liuyao88@mail.ustc.edu.cn

***Abstract*—Score normalization can improve speaker verification (SV) performance by adjusting the distribution of test scores to follow a normal distribution. In this paper, all of the imposter scores for the target speakers are first obtained from the normalization cohort; then, these scores are clustered by an unsupervised clustering algorithm, and Gaussian mixture models (GMMs) are used to fit the score distribution. The mean and the standard deviation of the Gaussian component with the maximum mean value is used in the SV score normalization method. Experiments are carried out on the NIST SRE 2016 test set and the VOiCES test set. Compared with conventional score normalization methods, the proposed method can effectively improve SV performance.**

***Index Terms*—speaker verification, score normalization, unsupervised clustering**

I. Introduction

Speaker verification system is used to verify a person's claimed identity by using voice characteristics. In such a typical two-category pattern-recognition task, the system makes decisions by comparing the test scores with a global detection threshold. In recent years, i-vector and x-vector based speaker verification systems have become mainstream methods due to their good performance [1, 2]. However, influenced by all kinds of variabilities, such as channel, language, duration, emotion and other factors, the score distribution of different speakers is highly stochastic, and a fixed threshold cannot achieve satisfactory performance.

Score normalization algorithms are often used to eliminate these kinds of randomness [3]. Traditionally, there are two basic methods: Z-norm and T-norm. In the Z-norm method, score normalization parameters are estimated from scores derived by scoring a set of imposter utterances through each target speaker model. In the T-norm method, the normalization parameters are estimated using scores derived at test time from a set of imposter speaker models [4]. Based on these two methods, other advanced methods, including the ZT-norm, the TZ-norm [5], the S-norm [6] and the KL-Tnorm [7] methods, are also proposed, and they have achieved obvious performance improvements in most of the state-of-the-art systems, such as the GMM-UBM [7], i-vector and x-vector based SV systems.

In the above-mentioned conventional score normalization methods, a set of imposter utterances are used to obtain normalization parameters, and they are always fixed for different speakers in the evaluation set. If the normalization cohort for the T-norm method or the Z-norm method are matched with the test conditions, these methods can achieve satisfactory results. However, normalization parameters estimated from the fixed cohort may be unsatisfactory in some cases. For example, there are no cross-sex trials or cross-language trials in the NIST SRE 2016 test set [8], and the normalization cohort consists of unlabeled utterances which means that we cannot use all scores obtained from the normalization cohort. To address this problem, adaptive cohort selection (ACS) algorithms are proposed to obtain normalization parameters using only part of the normalization cohort instead of the whole cohort [9, 10, 11]. The cohort can be adaptively selected at the model level [12] as well as at the score level [13]. The motivation behind ACS is that the utterances from the most competitive impostors are used to obtain the normalization parameters, the mean and the standard deviation of scores obtained from these utterances can reflect a more real distribution of a speaker model or a test utterance. Through adjusting the score distributions of different speaker models or test utterances to a similar distribution, the system performance improves significantly. In recent years, the most commonly used method is the Top-norm method in [13] in which we just select the top N scores for score normalization, but it still causes some problems. N is a hyper-parameter which is set experimentally using a development set, the system performance may vary intensely with different N. The distribution of top N scores is not Gaussian, and the estimated normalization parameters cannot represent the real distribution.

Inspired by the work mentioned above, we proposed a new type of ACS algorithm based on the unsupervised clustering algorithm. First, the K-means clustering algorithm [14] is applied to all of the scores. The scores belonging to the clusters with small mean values are discarded and will not be used for the following step. Then, an expectation-maximization (EM) algorithm [15] is applied, and GMMs are used to fit the distribution of the remaining scores. The parameters of the Gaussian component with the largest mean value are used for normalization. The method in this paper is experimentally verified on the evaluation set of the NIST SRE 2016 dataset and the VOiCES dataset.

The remainder of this paper is organized as follows. Section

2 introduces the mainstream score normalization algorithms. In section 3, we describe the proposed unsupervised clustering score normalization algorithm in detail. Section 4 presents the experimental setup and results. Finally, conclusions are given in section 5.

II. Score Normalization Technique

This section introduces several commonly used score normalization methods.

A. Conventional score normalization

The Z-norm and T-norm methods are the most widely used score normalization methods. Since they are similar, we only introduce Z-norm method in detail. For the m^{th} enrolled speaker model e_m, we can obtain a score set $s(e_m, t_l^*)$ using all the utterances $\{t_1^*, t_2^* ... t_L^*\}$ in the normalization cohort, where t_l^* is the l^{th} imposter utterance. We fit the distribution of these scores with a Gaussian distribution, and then the mean $\hat{\mu}$ and the standard deviation $\hat{\sigma}$ of the impostor scores can be obtained. These parameters can be used to normalize the actual test score $s(o)$ of e_m for the final decision.

$$s(o)_{norm} = \frac{s(o) - \hat{\mu}}{\hat{\sigma}} \tag{1}$$

Based on these two basic methods, S-norm method, where the scores normalized using T-norm method and the scores normalized using Z-norm method are averaged, is further proposed. All these methods use the whole normalization cohort to calculate the normalization parameters.

B. Adaptive score normalization

In adaptive score normalization methods, only part of the normalization cohort is selected to compute the mean and the standard deviation, and the selected utterances might change for every speaker. We use the ACS algorithm in the Top-norm method [14] as an example to illustrate this method. In the Top-norm method, the scores for the whole cohort are calculated as usual. However, only the top N scores are used to calculate the normalization parameters, as depicted in Figure 1. N is always determined by the development set.

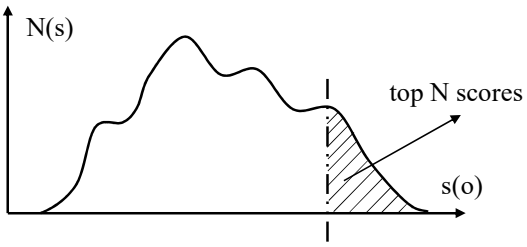


Fig. 1: Score selection in adaptive score normalization. The ordinate is the frequency of scores in a certain interval.

Since the mean and the standard deviation have changed, the formula of score normalization with ACS is as follows:

$$s(o)_{norm} = \frac{s(o) - \hat{\mu}_{topN}}{\hat{\sigma}_{topN}} \tag{2}$$

where $\hat{\mu}_{topN}$ and $\hat{\sigma}_{topN}$ are estimated from the top N scores. In most SV systems, score normalization with ACS can achieve better performance than conventional methods. In this paper, adaptive score normalization methods with the above-mentioned ACS algorithm will be used for comparison, and we call them top-N methods.

III. Clustering-Based Score Normalization

This section introduces an unsupervised clustering method to estimate these two parameters from scores that are obtained from the whole normalization cohort. The Z-norm method will be used as an example to illustrate the proposed method, and the T-norm method has a similar procedure. Similar to the Top-norm method, it will only use some high scores to estimate the normalization parameters. It contains two steps. The first step is a data-cleaning step in which the K-means algorithm is used and some scores are discarded, and the EM algorithm is applied to obtain the normalization parameters in the second step.

A. Data cleaning

In the NIST SRE 2016 evaluation set, there are no cross-sex trials or cross-language trials. These kinds of trials will exist when we use the whole normalization cohort for score normalization. Scores of these trials will be lower than the actual test scores obviously, so we only use the high scores obtained from competitive impostors. If we make full use of the information of these scores, the normalization parameters can have a positive impact on the final detection. In order to discard scores with small values adaptively, the K-means algorithm is used.

Let us suppose there are L scores $\{s(e_m, t_l^*), l \in [1, L]\}$ obtained from all imposter utterances, where e_m is the m^{th} speaker model in the enrollment set, and t_l^* is the l^{th} imposter utterance in the normalization cohort.

Algorithm1:K-means algorithm

1) Initialize the mean values of K clusters $\{\mu_1, \mu_2 \dots \mu_K\}$.
2) Classify each score into a cluster $C(\mu_k)$ based on minimum Euclidian distance:
   If $[s(e_m, t_l^*) - \mu_k]^2 \leq [s(e_m, t_l^*) - \mu_{k'}]^2, \forall k' \in [1, K]$ then $s(e_m, t_l^*) \in C(\mu_k)$
3) Update the mean value of each cluster:
   $\mu_k = \frac{1}{|C(\mu_k)|} \sum_{s(e_m, t_l^*) \in C(\mu_k)} s(e_m, t_l^*)$
   where $|C(\mu_k)|$ represents the number of scores belonging to cluster $C(\mu_k)$.
4) Repeat step 2) and 3) until the clusters converge.

The clusters with smaller mean value are discarded, and the scores in the top K' clusters are retained. In fact, if we use the mean and the standard deviation of the top one cluster for score normalization, it can also improve the system performance. Through our analysis, it is because the distribution of top one cluster is most similar to that of scores obtained under

the actual test conditions. However, using this step as a data cleaning step and using the information of top K' clusters to initialize parameters in the next step could achieve a further improvement.

B. Computing normalization parameters

We believe that the distribution of the remaining scores do not follow a single Gaussian distribution, and we want to find more reliable normalization parameters that can reflect the actual score distributions of speaker models or test utterances. GMMs are used to fit the score distributions, and the EM algorithm is used in the clustering step. We take the mean and the standard deviation of top K' clusters as the initial values of the GMMs, and the weight w_i is initialized according to the following formula:

$$w_i = \frac{|C(\mu_i)|}{|C|} \tag{3}$$

where $|C|$ represents the total number of remaining scores. Then the EM algorithm is applied to fit the score distribution.

Algorithm2:EM algorithm

1) Fix the parameter values $\{w_i, \mu_i, \sigma_i^2\}$ of the i^{th} Gaussian component, the posterior probability of score $s(e_m, t_l^*)$ can be calculated using Bayes' theorem:
$p(i|s(e_m, t_l^*)) = \frac{w_i N(s(e_m,t_l^*);m_i,s_i^2)}{\sum_j w_j N(s(e_m,t_l^*);m_j,s_j^2)}$
2) Update $\{w_i, \mu_i, \sigma_i^2\}$ using $p(i|s(e_m, t_l^*))$:
$w'_i = \frac{1}{|C|}\sum_{l=1}^{|C|} p(i|s(e_m, t_l^*))$
$\mu'_i = \frac{\sum_{l=1}^{|C|} s(e_m,t_l^*)p(i|s(e_m,t_l^*))}{\sum_{l=1}^{|C|} p(i|s(e_m,t_l^*))}$
$\sigma'^2_i = \frac{\sum_{l=1}^{|C|} p(i|s(e_m,t_l^*))(s(e_m,t_l^*)-\mu'_i)^2}{\sum_{l=1}^{|C|} p(i|s(e_m,t_l^*))}$

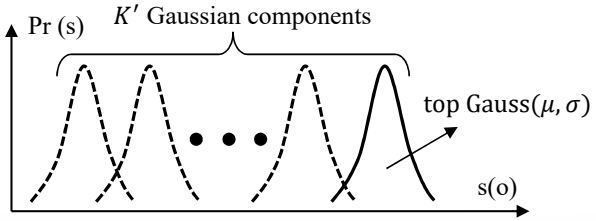


Fig. 2: Clustering using the EM algorithm. The ordinate is the probability of scores in a certain interval.

IV. Experiments and Discussion

A. Datasets

We carried out experiments on the core test of the NIST SRE 2016 data. There are approximately 2 million trials, with 37058 target and 1949462 non-target trials in the NIST SRE 2016 core test. The nominal durations of enrollment speech files are 60 s, while those of the test files vary from 10 to 60 s. The data from previous NIST SRE evaluations (2004-2010), Switchboard and Mix6 dataset are used as the training sets.

The NIST SRE 2016 corpus contains two major languages, Tagalog and Cantonese, which have never appeared in previous NIST SRE evaluation and training data. We use 2272 files from the NIST SRE 2016 development set as the normalization cohort, of which languages are matched with the evaluation set, but these files are unlabeled.

B. System description

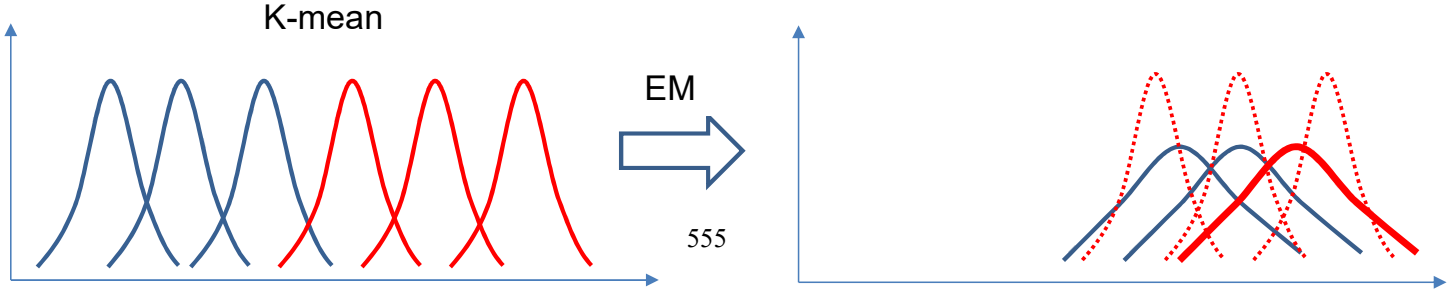
The experiments on the NIST SRE 2016 core data are based on the i-vector/PLDA framework. The whole process is mainly implemented with Kaldi open source code [16] while the PLDA is trained using in-house code. Mel-Frequency cepstral coefficient (MFCC) features with deltas and double deltas are extracted, which are 60-dimensional. A 3s sliding window is used for short-term mean and variance normalization. The voice activity detection (VAD) algorithm is used to remove silent frames. A gender-independent 2048-component GMM-UBM with diagonal covariance matrices is trained using unlabeled utterances of the NIST SRE 2016 development set. After the UBM is trained, a 600-dimensional total variability matrix is trained using the abovementioned training sets. After extracting the i-vectors, the training set and evaluation set are centred separately while the latter is centered using the mean of the unlabeled data. The i-vectors are reduced to 400 dimensions through the LDA algorithm. Since the prior of the G-PLDA model follows a Gaussian distribution, data whitening and length normalization are adopted before training the PLDA model. After preprocessing, the PLDA model is adopted as a backend classifier for speaker verification, where the sizes of speaker and channel matrices are 250 and 10, respectively.

In the stage of score normalization, three-quarters of utterances are randomly selected from the unlabeled data as the Z-norm set, and the remaining utterances are used as the T-norm set. Several mainstream score normalization methods are used for comparison.

C. Results

The equal error rate (EER), minimum error cost function (DCF^{min}) and actual error cost function (DCF^{act}) are used as evaluation metrics [17].

The experimental results are listed in Table 1. The system without any score normalization is marked as "baseline". "Z-norm", "T-norm" and "S-norm" represent the systems using all scores for normalization. When we use top N scores for computing the mean and the standard deviation, a prefix "top" is added for these system names. We use a prefix "GMM"



for the proposed score normalization method. The hyper-parameters in the score normalization methods are tuned using the development set of NIST SRE 2016 which also have two languages and some unlabeled data.

TABLE I: Results of different normalization methods

No.	Method	EER	DCF^{min}	DCF^{act}
0	Baseline	13.94	0.7716	0.9250
1	Z-norm	14.61	0.7871	0.8339
2	T-norm	14.27	0.7694	0.8133
3	S-norm	14.17	0.7685	0.8118
4	top Z-norm	14.23	0.7526	0.7993
5	top T-norm	13.93	0.7445	0.7732
6	top S-norm	13.72	0.7413	0.7701
7	GMM Z-norm	14.04	0.7411	0.7448
8	GMM T-norm	13.87	0.7292	0.7387
9	GMM S-norm	**13.69**	**0.7167**	**0.7214**

As shown in Table I, the GMM S-norm method performs best. Compared with the baseline, a relative 7.1% DCF^{min} and 22.0% DCF^{act} improvement are obtained. Additionally, the systems using adaptive cohort selection for score normalization can achieve better performance than the systems that do not use it.

D. The effect of hyper-parameters

There are some hyper-parameters in the top-N methods and the proposed score normalization methods. The system performance may vary with different hyper-parameters. The number of GMM components in the proposed unsupervised clustering method is a hyper-parameter. We list the results of the Z-norm method with different numbers of GMM components in Table II. The symbol "Δ" indicates the difference between the largest and the smallest value.

TABLE II: Results with different numbers of GMM components

Num	EER	DCF^{min}	DCF^{act}
4	14.13	0.7441	0.7525
5	**14.01**	0.7436	0.7463
6	14.04	**0.7411**	0.7448
7	14.08	0.7423	**0.7427**
8	14.16	0.7452	0.7502
Δ	0.15	0.0041	0.0098

TABLE III: Results with different top N numbers

Num	EER	DCF^{min}	DCF^{act}
150	14.23	**0.7526**	0.7993
200	14.35	0.7562	0.8024
250	**14.13**	0.7539	0.7965
300	14.28	0.7617	**0.7872**
350	14.35	0.7644	0.7923
400	14.42	0.7678	0.8171
Δ	0.29	0.0152	0.0299

N is the number of scores selected for the top-N methods, and we also list the results with different values of N in Table III. The top Z-norm method is used for comparison. From Table II and Table III, the performance fluctuation of the top-N methods is bigger than that of the proposed method, which means that the hyper-parameter of the proposed method has less effect on final detection performance. The hyper-parameters are usually tuned using a development set. However, there is no development set or the development set is mismatched with the evaluation set in most cases. The method proposed in this paper can solve this problem to some extent.

E. Bias of the estimated parameters

As mentioned above, score normalization method can achieve satisfactory results only if scores obtained from the normalization cohort is matched with the actual test conditions. We build a toy experiment here to calculate the bias between the mean of the normalization parameters and that of the test set. We first calculate the mean and standard deviation of impostor scores in the test set. The bias of the mean between the normalization cohort and test set is defined as follows:

$$\mu_{|bias|} = \frac{1}{M} \sum_{n=1}^{M} |\hat{\mu}_n - \mu_n| \tag{4}$$

where M is the total number of enrollment speakers (for Z-norm) or test utterances (for T-norm) in the test set; $\hat{\mu}_n$ is the mean used for score normalization; and μ_n is obtained from the test set. $\mu_{|bias|}$ reflects the bias between the normalization cohort and test set.

TABLE IV: Bias between the normalization cohort and the test set

Method	$\mu_{\|bias\|}$
Z-norm	93.6
top Z-norm	36.7
GMM Z-norm	**15.3**
T-norm	72.6
top T-norm	22.9
GMM T-norm	**17.4**

From the results in Table IV, we can see that the bias of the GMM score normalization is smaller than that of other methods. These results indicate that the proposed method can approximate the distribution of the test set better than other methods.

F. Additional experiments on VOiCES corpus

We also applied the proposed score normalization algorithm on the "VOiCES from a Distance Challenge 2019", and we develop the system for the fixed condition on two public datasets: VoxCeleb and SITW.

The VOiCES corpus is recorded in an acoustically challenging environment. The speech data contains much noise, reverberation, overlapping speech, laughter and acoustic artifacts, and the duration of speech utterances varies from 12 to 15 s. It consists of 20096 target trials and 3985782 non-target trials. We used VoxCeleb1 and VoxCeleb2 as the training set. The SITW corpus was used as a normalization cohort.

Our systems are based on the x-vector/PLDA and i-vector/PLDA frameworks. We build 22 subsystems using different i-vector and x-vector frontends. The PLDA algorithm is adopted as a backend classifier for all the i-vectors/x-vectors. The submitted systems are fused at score-level. A detailed description of our system can be found in [18].

Finally, we submit three systems. The scores of all subsystems are fused with equal weights for system 1. For system 2, the scores of each subsystem are normalized using the above mentioned unsupervised clustering score normalization. The score fusion weight of each subsystem is tuned in the development set. For system 3, the procedure of fusing subsystems is almost the same as that of system 2 except that score normalization is not applied.

As mentioned in [19], $\mathrm{C_{llr}}$ is an important metric to measure the quality of a system. For the purpose of analyzing how well a system is calibrated across all operating points, $\mathrm{C_{llr}}$ is defined as follows:

$$C_{llr} = \frac{1}{2 \times \log(2)} \times \left(\frac{\sum \log(1 + 1/s)}{N_{tar}} + \frac{\sum \log(1 + 1/s)}{N_{non}}\right) \tag{5}$$

where s is the likelihood ratio for a trial, and N_{tar} and N_{non} represent the number of target and non-target trials, respectively. The more reliable the identification system is, the lower the $\mathrm{C_{llr}}$ value is.

TABLE V: Results on test set of VOiCES from a Distance Challenge 2019

System	EER	$\mathrm{DCF^{min}}$	$\mathrm{DCF^{act}}$	$\mathrm{C_{llr}}$
1	6.82	**0.5088**	**0.5135**	1.8710
2	**6.71**	0.5201	0.5453	**0.4049**
3	6.79	0.5173	0.6234	1.4430

The final result on the test set is shown in Table V. The system that uses the proposed algorithm in this paper achieves competitive performance compared to other systems. We can see that the $\mathrm{C_{llr}}$ of system 2 improves significantly compared with system 1 and system 3 while the EER improves slightly. For fair comparison, we should only compare system 2 with system 3, because they have the same fusion strategy which is very important for the fusion systems. We can find that the $\mathrm{DCF^{act}}$ also improves. However, the $\mathrm{DCF^{act}}$ and $\mathrm{DCF^{min}}$ of system 2 are even worse than those of system 1. There are two reasons. On the one hand, the normalization cohort is not matched with the evaluation set in channel and some other factors. On the other hand, there are no rules in trials which does not like that in the NIST SRE 2016, and the proposed method cannot give full play to its advantage.

V. Conclusions

In this study, we employ clustering-based score normalization in the speaker verification system. The proposed method achieves an obvious performance improvement on the NIST SRE 2016 and the VOiCES from a Distance Challenge 2019. Further analysis shows that the proposed adaptive cohort selection algorithm can achieve stable performance.

Acknowledgment

This work was partially funded by the National Natural Science Foundation of China (Grant No. U1836219) and the National Key Research and Development Program of China (Grant No. 2016YFB100 1303).

References

[1] 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. 980C988, 2008.

[2] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," *2016 IEEE Spoken Language Technology Workshop (SLT).* IEEE, 2016, pp. 165C170.

[3] A. Rouzi, D. Wang, L. I. Lantian, F. Zheng, X. Zhang, and P. Jin, "Score domain speaking rate normalization for speaker recognition," *Journal of Tsinghua University,* 2018.

[4] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." *Odyssey,* vol. 14, 2010.

[5] H. Aronowitz, D. Irony, and D. Burshtein, "Modeling intraspeaker variability for speaker recognition," *Ninth European Conference on Speech Communication and Technology,* 2005.

[6] Y. Zigel and M. Wasserblat, "How to deal with multiple-targets in speaker identification systems?" *2006 IEEE Odyssey-The Speaker and Language Recognition Workshop.* IEEE, 2006, pp.1C7.

[7] D. E. Sturim, D. A. Reynolds, R. B. Dunn, and T. F. Quatieri, "Speaker verification using text-constrained gaussian mixture models," *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. 1. IEEE, 2002, pp. IC677.

[8] P. Matejka, O. Novotnỳ, O. Plchot, L. Burget, M. D. Sánchez, and J. Cernockỳ, "Analysis of score normalization in multilingual speaker recognition." *INTERSPEECH,* 2017, pp. 1567C1571.

[9] H. Khemiri and D. Petrovska-Delacretaz, "Cohort selection for text-dependent speaker verification score normalization," *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP).* IEEE, 2016, pp. 689C692.

[10] L. Skorkovská, Z.Zajíc, and L. Müller, "Comparison of score normalization methods applied to multi-label classification," *2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT).* IEEE, 2014, pp. 000 433C000 437.

[11] A. Swart and N. Brummer, "A generative model for score normalization in speaker recognition," *arXiv preprint arXiv:1709.09868,* 2017.

[12] D. E. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for tnorm in text-independent speaker verification," *Proceedings.(ICASSP05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.* vol. 1. IEEE, 2005, pp. IC741.

[13] D. Ramos-Castro, J. Fierrez-Aguilar, J. Gonzalez-Rodriguez, and J. Ortega-Garcia, "Speaker verification using speaker-and testdependent fast score normalization," *Pattern recognition letters,* vol. 28, no. 1, pp. 90C98, 2007.

[14] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society,* vol. 28, no. 1, pp. 100C108, 1979.

[15] 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 (Methodological),* vol. 39, no. 1, pp. 1C22, 1977.

[16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[17] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2016 nist speaker recognition evaluation." *Interspeech* 2017, pp. 1353C1357.

[18] W. G. Lanhua You, Bin Gu, "Ustcspeech system for voices from a distance challenge 2019," *arXiv preprint arXiv:1903.12428,* 2019.

[19] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, A. Lawson, and M. A. Barrios, "The voices from a distance challenge 2019 evaluation plan," *arXiv preprint arXiv:1902.10828,* 2019.