# SIMPLIFIED VTS-BASED I-VECTOR EXTRACTION IN NOISE-ROBUST SPEAKER RECOGNITION

*Yun Lei* *Mitchell McLaren* *Luciana Ferrer* *Nicolas Scheffer*

Speech Technology and Research Laboratory, SRI International, California, USA

{yunlei,mitch,lferrer,scheffer}@speech.sri.com

## ABSTRACT

A vector taylor series (VTS) based i-vector extractor was recently proposed for noise-robust speaker recognition by extracting synthesized clean i-vectors to be used in the standard system back-end. This approach brings significant improvements in accuracy for noisy speech conditions. However, this approach incurred such a large computational expense that using the state-of-the-art model size or evaluating large scale evaluations was impractical. In this work, we propose an efficient simplification scheme, named sVTS, in order to show that the VTS approach gives improvements in large scale applications compared to state-of-the-art systems. In contrast to VTS, sVTS generates normalized Baum-Welch statistics and uses the standard i-vector model, making it straightforward to employ on the state-of-the-art i-vector speaker recognition system. Results presented on both the PRISM and the large NIST SRE'12 corpora show that using sVTS i-vectors provides significant improvements in the noisy conditions, and that our proposed simplification result in only a slight degradation with respect to the original VTS approach.

**Index Terms**: speaker recognition, Vector Taylor Series, i-vector, noisy speaker verification, noise compensation

## 1. INTRODUCTION

Recently, the state-of-the-art in speaker verification has seen significant improvements in accuracy from the successful application of the i-vector extraction paradigm [1], along with a Bayesian back-end (such as probabilistic linear discriminant analysis (PLDA) [2, 3, 4]). In this framework, each speech utterance is projected into a single low-dimensional vector – referred to as i-vector – of a few hundred dimensions, and a PLDA model is then used to compare i-vectors from different utterances to produce verification scores.

The approach proposed in this work tackles the challenge of robustness to noisy speech for speaker verification systems. While current state-of-the-art systems achieve very high accuracy on clean speech, the degradation incurred by noise is still a challenge. In [5], we have successfully proposed a robust strategy to compensate for such degradation by using multi-style training for the PLDA back-end. While significant improvements were obtained, there is still an order of magnitude difference in accuracy between clean and noisy speech.

In our previous work [6], we proposed to tackle the problem at an earlier stage, where the i-vector extractor explicitly takes into account the modeling of noise in the speech data using the VTS approximation. VTS is used to model nonlinear distortions in the mel-cepstral domain caused by both additive and convolutive noise. In automatic speech recognition (ASR), VTS is used to synthesize an acoustic model of noisy speech from a given clean speech model and from estimated noise distributions [7, 8, 9].

In contrast to ASR, we use the VTS approach in a somewhat opposite manner since our goal is to obtain a clean version of an i-vector. We effectively model the relation between the "clean" i-vector and the corresponding "noisy" GMM and compute the "clean" i-vector directly by fitting the corresponding noisy GMM to a given noisy speech segment. Although this approach provides significant improvements on noisy data, this new framework results in a high computational burden and memory usage. It is therefore impractical to use similar model size as in state-of-the-art systems, which hinders the possibility of making comparisons on large scale evaluations.

In this work, we propose an efficient simplification of the the VTS framework for speaker recognition: sVTS. We first validate this new model by a comparison with the original VTS and a baseline system on the noisy PRISM corpus. We then show results and benefits of our approach on the large NIST SRE 2012 evaluation using standard model sizes as found in the literature. The main benefit of our proposed approach is that it works at the sufficient statistics level, enabling the use of the standard equations for i-vector extraction (and i-vector model training) avoiding most of the computational burden introduced by the original approach.

## 2. STANDARD I-VECTOR EXTRACTION

In the standard i-vector framework, (clean) speech frames $\mathbf{x}^{(i)}$ from the $i$-th speech segment are assumed to be generated by the following distribution:

$$\mathbf{x}^{(i)} \sim \sum_m \pi_m N(\boldsymbol{\mu}_m + \mathbf{T}_m \boldsymbol{\omega}^{(i)}, \boldsymbol{\Sigma}_m), \quad (1)$$

where $\boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_m$ and $\pi_m$ are means, covariance matrices and weights of the universal background model (UBM), matrix $\mathbf{T}_m$ spans a low-rank subspace (referred to as the total variability subspace) by which GMM means are adapted to a particular speech segment, and $\boldsymbol{\omega}^{(i)}$ is a segment-specific low-dimensional vector with standard Gaussian prior distribution. Given a speech segment, the i-vector is computed using Equation (2) below as the maximum a posteriori (MAP) point estimate of the vector $\boldsymbol{\omega}^{(i)}$.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D10PC20024. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA or its Contracting Agent, the U.S. Department of the Interior, National Business Center, Acquisition & Property Management Division, Southwest Branch. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. "A" (Approved for Public Release, Distribution Unlimited). We thank Lukas Burget for many useful discussions during the development of this work.

The subspace $\mathbf{T}_m$ can be trained using the expectation maximization (EM) algorithm [10]. In the *E step*, the posterior distribution of $\boldsymbol{\omega}^{(i)}$ is Gaussian with mean and covariance matrices given by:

$$\langle \boldsymbol{\omega}^{(i)} \rangle = \mathbf{L}^{(i)} \sum_m \tilde{\mathbf{T}}_m^T \tilde{\mathbf{f}}_m^{(i)} \quad (2)$$

$$\mathbf{L}^{(i)} = (I + \sum_m \gamma_m^{(i)} \tilde{\mathbf{T}}_m^T \tilde{\mathbf{T}}_m)^{-1} \quad (3)$$

where the matrix $\tilde{\mathbf{T}}_m$ relates to the matrix $\mathbf{T}_m$ from eq. (1) as $\mathbf{T}_m = \mathbf{P}_m \tilde{\mathbf{T}}_m$, and $\mathbf{P}_m$ is a lower triangular matrix obtained form the Cholesky decomposition of $\boldsymbol{\Sigma}_m = \mathbf{P}_m \mathbf{P}_m^T$, and

$$\gamma_m^{(i)} = \sum_t \gamma_{mt}^{(i)} \quad (4)$$

$$\tilde{\mathbf{f}}_m^{(i)} = \mathbf{P}_m^{-1} \sum_t \gamma_{mt}^{(i)} (\mathbf{x}_t^{(i)} - \boldsymbol{\mu}_m) \quad (5)$$

are the zero order and first order *whitened* sufficient statistics precollected using UBM. The first order statistic whitening (i.e. $\boldsymbol{\mu}_m^{(i)}$ subtraction and multiplication by $\mathbf{P}_m^{-1}$) not only leads to more efficient implementation (i.e. simpler formulas (2)), but it will also play an important role in the simplified VTS approach proposed in section 4.

In the *M step*, the matrices $\tilde{\mathbf{T}}_m$ are updated as:

$$\tilde{\mathbf{T}}_m = \sum_i \tilde{\mathbf{f}}_m^{(i)} \langle \boldsymbol{\omega}^{(i)} \rangle^T \left( \sum_i \gamma_m^{(i)} \left( \mathbf{L}^{(i)} + \langle \boldsymbol{\omega}^{(i)} \boldsymbol{\omega}^{(i)} \rangle^T \right) \right)^{-1} \quad (6)$$

## 3. VTS-BASED I-VECTOR EXTRACTION

This section describes the original idea of applying the VTS approximation to the model for noise robust i-vector extraction as introduced in [6]. The VTS-based i-vector extraction is a two-step process: 1) the UBM is first adapted to the additive and convolutive noise of a speech segment, and 2) the noise-compensated i-vector is then extracted based on the sufficient statistics collected from the adapted UBM. We first describe the UBM noise adaptation along with the foundation of the VTS approximation, originally proposed for noise robust ASR in [8]. We then derive the noise robust i-vector extraction model using similar approximations.

### 3.1. Adaptating UBM to noise

In the mel-frequency cepstrum (MFCC) domain, the feature vector for a noisy speech frame $\mathbf{y}$ can be modeled as

$$\mathbf{y} = \mathbf{x} + \mathbf{h} + g(\mathbf{n} - \mathbf{x} - \mathbf{h}), \quad (7)$$

where $\mathbf{y}, \mathbf{x}, \mathbf{h}, \mathbf{n}$ are the cepstrum vectors corresponding to the noisy speech, clean speech, channel, and additive noise, respectively. The nonlinear function $g$ is

$$g(\mathbf{n} - \mathbf{x} - \mathbf{h}) = \mathbf{C} \log(1 + \exp(\mathbf{C}^\dagger (\mathbf{n} - \mathbf{x} - \mathbf{h}))), \quad (8)$$

where $\mathbf{C}$ is the discrete cosine transform (DCT) matrix and $\mathbf{C}^\dagger$ is its pseudo-inverse. Assuming Gaussian distributions for both additive and convolutive noise, the mean vector of the $m$-th component of the noise-adapted UBM can be approximated using a VTS expansion at $(\boldsymbol{\mu}_{x_m 0}, \boldsymbol{\mu}_{n0}, \boldsymbol{\mu}_{h0})$ as

$$\begin{aligned} \boldsymbol{\mu}_{y_m} \approx\ & \boldsymbol{\mu}_{x_m 0} + \boldsymbol{\mu}_{h0} + g(\boldsymbol{\mu}_{n0} - \boldsymbol{\mu}_{x_m 0} - \boldsymbol{\mu}_{h0}) \\ & + \mathbf{G}_m(\boldsymbol{\mu}_{x_m} - \boldsymbol{\mu}_{x_m 0}) + \mathbf{G}_m(\boldsymbol{\mu}_h - \boldsymbol{\mu}_{h0}) \\ & + \mathbf{F}_m(\boldsymbol{\mu}_n - \boldsymbol{\mu}_{n0}), \end{aligned} \quad (9)$$

where $\boldsymbol{\mu}_{x_m}$ is the mean of the corresponding component in the clean UBM, and $\boldsymbol{\mu}_n$ and $\boldsymbol{\mu}_h$ are the means of the additive and convolutive noise distributions, respectively. The matrices $\mathbf{G}_m$ and $\mathbf{F}_m$ are defined as:

$$\begin{aligned} \mathbf{G}_m &= \mathbf{C} \cdot \mathrm{diag}\left( \frac{1}{1 + \exp(\mathbf{C}^\dagger (\boldsymbol{\mu}_{n0} - \boldsymbol{\mu}_{x_m 0} - \boldsymbol{\mu}_{h0}))} \right) \cdot \mathbf{C}^\dagger \\ \mathbf{F}_m &= \mathbf{I} - \mathbf{G}_m. \end{aligned} \quad (10)$$

To synthesize the noisy UBM, the VTS expansion is done at the point $(\boldsymbol{\mu}_{x_m 0} = \boldsymbol{\mu}_{x_m}, \boldsymbol{\mu}_{n0} = \boldsymbol{\mu}_n, \boldsymbol{\mu}_{h0} = \boldsymbol{\mu}_h)$, which reduces (9) to

$$\boldsymbol{\mu}_{y_m 0} \approx \boldsymbol{\mu}_{x_m 0} + \boldsymbol{\mu}_{h0} + g(\boldsymbol{\mu}_{n0} - \boldsymbol{\mu}_{x_m 0} - \boldsymbol{\mu}_{h0}). \quad (11)$$

The noise-adapted covariance matrix can be approximated as

$$\boldsymbol{\Sigma}_{y_m} \approx \mathbf{G}_m \boldsymbol{\Sigma}_{x_m} \mathbf{G}_m^T + \mathbf{F}_m \boldsymbol{\Sigma}_n \mathbf{F}_m^T, \quad (12)$$

where $\boldsymbol{\Sigma}_{x_m}$ is the covariance matrix of the $m$-th Gaussian component from the clean UBM, $\boldsymbol{\Sigma}_n$ is the additive noise covariance matrix and $\boldsymbol{\Sigma}_h$ is set to zero since the channel is usually considered to be fixed. A similar adaptation can be derived for the first-order and second-order derivatives of the MFCC features [6]. The noise distribution parameters $\boldsymbol{\mu}_n$, $\boldsymbol{\mu}_h$ and $\boldsymbol{\Sigma}_n$ can be estimated directly from the noisy speech segments [8, 6].

### 3.2. Noise-compensated i-vector extraction

By incorporating the VTS approximation (9) and (12) into the i-vector extraction model (1), the model for the noisy features $\mathbf{y}^{(i)}$ is given by (see [6])

$$\mathbf{y}^{(i)} \sim \sum_m \pi_m N(\boldsymbol{\mu}_{y_m 0}^{(i)} + \mathbf{G}_m^{(i)} \mathbf{T}_m \boldsymbol{\omega}^{(i)}, \boldsymbol{\Sigma}_{y_m}^{(i)}). \quad (13)$$

According to [6], the resulting i-vector $\boldsymbol{\omega}^{(i)}$ and EM algorithm for training the subspace $\mathbf{T}_m$ can be derived as follows: In the *E step*, the posterior distribution of $\boldsymbol{\omega}^{(i)}$ in (13) is Gaussian with mean and covariance matrices given by:

$$\langle \boldsymbol{\omega}^{(i)} \rangle = \mathbf{L}^{(i)} \sum_m \mathbf{T}_m^T (\hat{\boldsymbol{\Sigma}}_{y_m}^{(i)})^{-1} \mathbf{f}_{y_m}^{(i)}, \quad (14)$$

$$\mathbf{L}^{(i)} = (I + \sum_m \gamma_m^{(i)} \mathbf{T}_m^T (\hat{\boldsymbol{\Sigma}}_{y_m}^{(i)})^{-1} \mathbf{T}_m)^{-1}, \quad (15)$$

where statistics $\mathbf{f}_{y_m}^{(i)}$ and $(\hat{\boldsymbol{\Sigma}}_{y_m}^{(i)})^{-1}$ are pre-collected from a noisy speech segment using the noise-adapted UBM as

$$\begin{aligned} \mathbf{f}_{y_m}^{(i)} &= \sum_t \gamma_{mt}^{(i)} (\mathbf{G}_m^{(i)})^{-1} (\mathbf{y}_t^{(i)} - \boldsymbol{\mu}_{y_m}^{(i)}) \\ (\hat{\boldsymbol{\Sigma}}_{y_m}^{(i)})^{-1} &= (\mathbf{G}_m^{(i)})^T (\boldsymbol{\Sigma}_{y_m}^{(i)})^{-1} \mathbf{G}_m^{(i)}. \end{aligned} \quad (16)$$

In the *M step*, the matrix $\mathbf{T}_m$ can be updated as

$$\begin{aligned} \mathrm{vec}(\mathbf{T}_m) = & \left( \sum_i \gamma_{ym}^{(i)} \left( \mathbf{L}^{(i)} + \langle \boldsymbol{\omega}^{(i)} \boldsymbol{\omega}^{(i)} \rangle^T \right) \otimes (\hat{\boldsymbol{\Sigma}}_{y_m}^{(i)})^{-1} \right)^{-1} \\ & \times \mathrm{vec} \sum_i (\hat{\boldsymbol{\Sigma}}_{y_m}^{(i)})^{-1} \mathbf{f}_{ym}^{(i)} \langle \boldsymbol{\omega}^{(i)} \rangle^T \end{aligned} \quad (17)$$

where $\otimes$ is the Kronecker product and vec is an operator which creates a column vector from a matrix by stacking its columns.

## 4. SIMPLIFICATION OF VTS-BASED I-VECTOR EXTRACTION

An efficient implementation of standard i-vector extraction can benefit from pre-computing the quadratic terms $\tilde{\mathbf{T}}_m^T \tilde{\mathbf{T}}_m$, which slows down the i-vector extraction by an order of magnitude when they are evaluated on-the-fly. The *E step* for $\tilde{\mathbf{T}}_m$ training can benefit from the whitening of the sufficient statistics as shown in equation (5). Neither of these two optimization is possible in the case of the VTS-based i-vector extraction because of the segment dependent term $\boldsymbol{\Sigma}_{y_m}^{(i)}$. More importantly, in the *M step*, the Kronecker product and the large matrix inverse in equation (17) is several orders of magnitude more computationally and memory demanding than the calculations in the equation (6). For these reasons, it is impractical to use the noise-compensated i-vector extraction approach for models of large size or to run large scale evaluations.

In this work, we propose an efficient approximation, referred to as simplified VTS (sVTS), for noise-compensated i-vector extraction, which eliminates most of the computational burden. First, for each speech segment (training, enrollment or test), we adapt the UBM to the noise in the segment in the same way as described in section 3.1. Next, the sufficient statistics are collected using the equations for standard i-vector extraction (4) and (5), except that the segment-dependent means, $\boldsymbol{\mu}_{y_m}^{(i)}$, from the synthesized noisy UBM are used in place of the original means $\boldsymbol{\mu}_m$, and similarly, the matrices $\mathbf{P}_m$ are replaced by segment-dependent lower triangular matrices $\mathbf{P}_m^{(i)}$, which are obtained from the Cholesky decompositions of the noisy covariance matrices $\boldsymbol{\Sigma}_{y_m}^{(i)}$. The occupation counts $\gamma_{mt}^{(i)}$ in equations (4) and (5) are also collected using the synthesized noisy UBM. These statistics are then used for i-vector extraction and training using the standard formula (2), (3) and (6). As a result, the only difference from the standard i-vector extraction scheme is that the sufficient statistics are collected and *whitened* using the VTS synthesized noisy UBM. Such whitening can be seen as a transformation of the noisy sufficient statistics to a "clean canonical" domain. Therefore, sVTS can be seen as a noise compensation technique operating in the domain of sufficient statistics, while our former VTS approach is a model-domain compensation technique. The following experiments show that the sVTS approach can preserve most of the improvements obtained with the original VTS-based i-vector extraction model.

## 5. EXPERIMENTS AND RESULTS

We first compare the new sVTS approach to the original VTS using a small model[1] on the PRISM noise set [11]. Once sVTS is validated, we show results and benefits of sVTS on the standard NIST SRE'12 noisy conditions, where the test segments include additive noise, and trials are channel-mismatched.

### 5.1. Evaluation on PRISM set using small models

The frontend for all systems is comprised of 20 MFCC coefficients (including C0), augmented with first-order and second-order derivatives. In addition, the baseline system (i.e. standard i-vector framework) applies mean and variance normalization (MVN) to the MFCC features as MVN has been widely used to compensate for additive and convolutive noise in speaker recognition.

[1] A small model is used for this comparison since it is computationally impractical to apply the standard VTS on large models.

A 512-component diagonal covariance UBM is trained in a gender-dependent fashion on NIST telephone data from the 2004 and 2005 speaker recognition evaluations (SRE). An i-vector extractor of dimension 400 is then trained on a set taken from NIST SRE '04, '05, '06, and Switchboard II parts 2 and 3. The dimensionality of i-vectors is further reduced to 200 by LDA, followed by length normalization and PLDA, trained on the same data set.

Results are shown on a part of the PRISM set described in [5, 11], where noisy speech samples are added to the training, enrollment, and test sets at different signal-to-noise ratios (SNR) of 20dB, 15dB, and 8dB. Different noisy samples are used for the training, enrollment and test sets. We report performance in terms of the detection cost function (DCF) and equal error rate (EER) at each SNR level. The DCF effective prior used is the one from the NIST SRE'10 [12] evaluation, denoted as DCF10.

**Table 1**. *DCF10 and EER performance of the baseline and VTS systems compared to the proposed sVTS approach where both clean and multistyle PLDA back-ends were used. The sVTS system significantly outperforms the baseline system but slightly underperforms the VTS system in low SNR conditions.*

a. *Clean* PLDA back-end (DCF10 / EER(%))

| Eval. cond. | MVN | VTS | sVTS |
|---|---|---|---|
| SNR=8dB | 0.98 / 15.53 | 0.64 / 5.25 | 0.77 / 5.25 |
| SNR=15dB | 0.66 / 4.09 | 0.27 / 1.76 | 0.33 / 1.85 |
| SNR=20dB | 0.35 / 1.94 | 0.18 / 1.23 | 0.22 / 1.52 |
| Clean | 0.08 / 0.53 | 0.15 / 0.82 | 0.11 / 0.82 |

b. *Multistyle* PLDA back-end (DCF10 / EER(%))

| Eval. cond. | MVN | VTS | sVTS |
|---|---|---|---|
| SNR=8dB | 0.81 / 5.98 | 0.48 / 3.29 | 0.55 / 3.32 |
| SNR=15dB | 0.44 / 2.16 | 0.23 / 1.35 | 0.30 / 1.44 |
| SNR=20dB | 0.26 / 1.33 | 0.17 / 1.03 | 0.19 / 1.21 |
| Clean | 0.09 / 0.40 | 0.15 / 0.64 | 0.11 / 0.62 |

Table 1 presents the performance of the baseline (MVN), VTS, and sVTS systems at different SNRs. Two PLDA back-ends are evaluated: *Clean*, where the PLDA model is trained exclusively on clean data, and *Multistyle*, where the PLDA model is trained on clean and noisy data as proposed in [5]. From the results, we observe large improvements offered by VTS over MVN in noisy conditions using a relatively small model size, even when using multi-style PLDA. These improvements over the baseline system are largely maintained using our sVTS approach, especially in low SNR conditions. Although a slight degradation is observed when using sVTS over VTS, the benefit of having no additional computation required after the UBM adaptation makes the approach scalable to larger model size. Though not shown in the table, similar improvements can be observed when SNR levels are mismatched for enrollment and test.

### 5.2. Evaluation on NIST SRE12 noisy conditions

In contrast to previous years, three new noisy conditions were introduced in NIST SRE12 [13]: interview speech with added noise (C3), telephone speech with added noise (C4), and telephone speech collected under noisy conditions (C5). For this work, we only present results on C3 and C4 (C5 was discarded as the effect of additive noise is marginal).

Compared to the PRISM noisy set, these conditions present several differences. First, the enrollment data for all speakers can be used for system training. Second, all enrollment sessions are in

clean conditions, and finally, a speaker model is derived from multiple enrollment sessions with both telephone and microphone channels. These differences make it cumbersome to use a VTS approach for noise compensation directly since we showed that they affect the performance in clean conditions. To mitigate this effect, we propose to combine the original statistics as well as the VTS compensated statistics during PLDA training and system evaluation.

To ease the experimental burden, results are shown on the female trials only. A 2048-component, gender-dependent, GMM with diagonal covariances is used along with 600 dimensional i-vectors further reduced to 200 by LDA followed by length normalization and PLDA. The development set and model training of the system is described in [14] and only clean enrollment data is used to model speakers in the evaluation. As before, the baseline system uses MVN but employs the exact same model configuration described above.

A single i-vector extractor, named *Comb*, is trained using both original and VTS-compensated statistics for the NIST SRE12 noisy conditions without MVN. During evaluation, we use only the original statistics to extract the i-vectors for the clean enrollment files, while the i-vectors from noisy test samples are extracted from the VTS-compensated statistics. For comparison, the performance from another system called *Raw* is shown, which has the same setup as the baseline MVN system but omits the MVN process.

#### *5.2.1. Evaluation with clean PLDA back-ends*

We first evaluate the three systems with a clean backend trained without any noisy data. The *Comb* system used both the original and the compensated i-vectors for every segment of clean data to train the back-end. This reduces the mismatch between the original i-vectors corresponding to the clean enrollment samples and compensated i-vectors corresponding to the noisy test samples. The approach also alleviates the mismatch between uncompensated enrollment and compensated test i-vectors.

**Table 2**. *minDCF (Ptar=0.001 / Ptar=0.01) performance on NIST SRE12 conditions for the* MVN *baseline and* Raw *systems compared to the proposed* Comb *approach where sVTS compensated i-vectors are used in the clean PLDA back-end. The* Comb *system consistently outperforms the other two systems.*

| Condition | MVN | Raw | Comb |
|---|---|---|---|
| C3 | .259 / .138 | .304 / .173 | .237 / .129 |
| C4 | .475 / .293 | .531 / .323 | .468 / .280 |

Table 2 presents the minDCF, with $P_{tar} = 0.001$ and $P_{tar} = 0.01$ as defined in [13], of the *MVN* baseline, *Raw*, and *Comb* systems with two different clean PLDA back-ends for the NIST SRE12 extended C3 and C4 noisy tasks. The *Raw* and *Comb* systems mainly differ in the statistics computation of the noisy segments (statistics for the clean samples are identical). Hence, the significant improvement observed from the *Comb* system compared to the *Raw* system is due to the proposed sVTS compensation applied on the noisy segments. The combination of original and compensated statistics used in the *Comb* system smooths the mismatch between i-vectors obtained for the clean and the noisy segments, which results in a system that outperforms the *MVN* baseline. If the backend is trained only on the original i-vectors, the performance of the *Comb* system degrades significantly. The improvements achieved with the clean back-end confirm the benefit of the sVTS compensation approach.

#### *5.2.2. Evaluation with multi-style PLDA back-ends*

For the multi-style PLDA back-end, sVTS-compensated i-vectors are extracted for the noisy data. In this case. it is not necessary to duplicate the clean data by computing both original and compensated i-vectors. To clarify the difference between clean and multi-style back-ends in the *Comb* system, table 3 presents the i-vector selection strategy for the *Comb* system. As was the case for the clean backend, Table 4 shows that the proposed *Comb* system outperforms both the *MVN* and *Raw* systems. The improvements from the *Comb* system show the complementarity of the sVTS compensation in the frontend and the multi-style training in the back-end.

**Table 3**. *i-vector selection strategy for different training regimes of the PLDA back-end in the* Comb *system.*

| PLDA back-end | data type | original | compensated |
|---|---|---|---|
| Clean | clean | × | × |
| Multistyle | clean | × | |
| | noisy | | × |

**Table 4**. *The min DCF (Ptar=0.001 / Ptar=0.01) performance on NIST SRE12 data for* MVN *baseline and* Raw *systems compared to the proposed* Comb *approach where sVTS compensated i-vectors are used in the clean PLDA backends. The* Comb *system outperforms the other two systems consistently.*

| Condition | MVN | Raw | Comb |
|---|---|---|---|
| C3 | .202 / .100 | .233 / .125 | .178 / .090 |
| C4 | .405 / .226 | .453 / .257 | .388 / .230 |

## 6. CONCLUSIONS

In this work, we aim at showing the benefit of the new VTS approach for speaker recognition by running evaluations on large data sets and using a model size typically found in the state-of-the-art. Given the computational burden of our original VTS solution, we propose an efficient approximation, sVTS, which collects sufficient statistics and whitens them using the VTS-synthesized UBM. As demonstrated by results on the PRISM corpus, this new approach preserves most of the improvements obtained from the original VTS approach but similar computation burden as the standard i-vector model. Moreover, sVTS can easily be evaluated on large data sets, such as the noisy conditions of NIST SRE 2012. In this scenario, we still find that our VTS approach outperforms the state-of-the-art on both extended conditions C3 and C4 which contain noisy speech. For a successful application of the sVTS approach on this corpora, we proposed the design of a system that combines both VTS and the standard approach to counteract the degradation incurred by VTS on clean data. Future research will include the study of VTS system performance under other types of noise and degradations like convolutive noise, reverberant speech, and so on.

## 7. REFERENCES


[1] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. ASLP*, vol. 19, pp. 788–798, May 2010.

[2] S.J.D. Prince, "Probabilistic linear discriminant analysis for inferences about identity," in *ICCV-11th*. IEEE, 2007, pp. 1–8.

[3] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey 2010-The Speaker and Language Recognition Workshop*. IEEE, 2010.

[4] D. Garcia-Romero and C.Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech-2011*, August 2011, pp. 249–252.

[5] Y. Lei, L. Burget, L. Ferrer, M. Graciarena, and N. Scheffer, "Towards noise-robust speaker recognition using probabilistic linear discriminant analysis," in *ICASSP-2012*. IEEE, March 2012, pp. 4253–4256.

[6] Y. Lei, L. Burget, and N. Scheffer, "A noise robust i-vector extractor using vector Taylor series for speaker recognition," in *ICASSP-2013*. IEEE, May 2013.

[7] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "Hmm adaptation using vector Taylor series for noisy speech recognition," in *ICSLP*, 2000, vol. 3, pp. 229–232.

[8] O. Kalinli, M. Seltzer, J. Droppo, and A Acero, "Noise adaptive training for robust automatic speech recognition," *IEEE Trans. ASLP*, vol. 18, pp. 1889–1901, Nov. 2010.

[9] H Liao, *Uncertainty Decoding for Noise Robust Speech Recognition*, PhD dissertation, University of Cambridge, Sept. 2007.

[10] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of inter-speaker variability in speaker verification," *IEEE Trans. ASLP*, vol. 16, pp. 980–988, July 2008.

[11] L. Ferrer, H. Bratt, L. Burget, J. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, and N. Scheffer, "Promoting robustness for speaker modeling in the community: the PRISM evaluation set," in *Proceedings of NIST 2011 Workshop*, 2011.

[12] "NIST SRE10 evaluation plan," http://www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf.

[13] "NIST SRE12 evaluation plan," http://www.nist.gov/itl/iad/mig/upload/NIST_SRE12_evalplanv17-r1.pdf.

[14] L. Ferrer, M. McLaren, N. Scheffer, Y. Lei, M Graciarena, and V. Mitra, "A noise-robust system for NIST 2012 speaker recognition evaluation," in *Interspeech-2013*, 2013.