Fast i-vector denoising using MAP estimation and a noise distributions database for robust speaker recognition

Waad Ben Kheder, Driss Matrouf, Pierre-Michel Bousquet Bousquet, Jean-François Bonastre, Moez Ajili

Fast i-vector denoising using MAP estimation and a noise distributions database for robust speaker recognition

Waad Ben Kheder, Driss Matrouf, Pierre-Michel Bousquet
Jean-François Bonastre and Moez Ajili

LIA, University of Avignon, France

Abstract

Once the i-vector paradigm has been introduced in the field of speaker recognition, many techniques have been proposed to deal with additive noise within this framework. Due to the complexity of its effect in the i-vector space, a lot of effort has been put into dealing with noise in other domains (speech enhancement, feature compensation, robust i-vector extraction and robust scoring). As far as we know, there was no serious attempt to handle the noise problem directly in the i-vector space without relying on data distributions computed on a prior domain. The aim of this paper is twofold. First, it proposes a full-covariance Gaussian modeling of the clean i-vectors and noise distribution in the i-vector space and introduces a technique to estimate a clean i-vector given the noisy version and the noise density function using the MAP approach. Based on NIST data, we show that it is possible to improve by up to 60% the baseline system performance. Second, in order to make this algorithm usable in a real application and reduce the computational time needed by i-MAP, we propose an extension that requires building a noise distribution database in the i-vector space in an off-line step and using it later in the test phase. We show that it is possible to achieve comparable results using this approach (up to 57% of relative EER improvement) with a sufficiently large noise distribution database.

Keywords: i-vectors, MAP adaptation, speaker recognition, additive noise.

1. Introduction

Recent work on the robustness of speaker recognition systems based on i-vectors technology has been carried out at different levels in order to track and compensate the additive noise effect without altering the speaker-related information. Additive noise has always been one of the most important problems in speaker recognition research and dealing with it generally falls into one of four categories: speech enhancement, feature compensation, robust modeling or score compensation. We will not discuss the latter here as it does not deal directly with additive noise.

At signal level, [1] proved that spectral and wavelet-based speech enhancement techniques do not perform consistently when used as a pre-processing block in a standard speaker recognition system even if the resultant speech quality increases. It was further shown in [2] that these algorithms might either enhance or degrade the recognition performance depending on the noise type and the SNR level. The speaker-related information has been proven to be vulnerable and hard to handle in this domain due to the natural complexity and redundancy in the speech signal which led to the development of other techniques based on different domains.

At feature level, [3] carried out an extensive comparison of several spectrum estimation methods under additive noise contamination and found that the best spectrum estimator was related to the noise type and level. Recent work [4, 5], based on vector Taylor series (VTS) then developed using "unscented transforms" [6] tried to model non-linear distortions in the cepstral domain based on a non-linear noise model in order to relate clean and noisy cepstral coefficients and help estimate a "cleaned-up" version of i-vectors. Despite its efficiency, this model remains very rigid due to its complexity and not easily extensible. In such a technique, adding a normalization step or changing the parameters used could involve rewriting the whole technique. On another level, a set of stochastic techniques originally introduced for robust speech recognition such as RATZ [7], SPLICE [8], SSM [9] and TRAJMAP [10] have lately been investigated for

speaker recognition [11]. In these techniques, the effect of noise is represented by additive terms in the mean vectors and covariance matrices of clean speech GMMs. Although some of these algorithms achieve very good results (SSM and TRAJMAP), a priori knowledge about the test environment is assumed and stereo training data is required.

On the model level, prior knowledge about the test environment is used in the form of a statistical model of the noise or a reliable estimate of the noise distribution. Model compensation techniques are usually superior to their feature-level counterparts since they can capture the uncertainty caused by the noise statistics [12]. The parallel model combination (PMC) was first introduced in speech recognition technology [13] before being adapted to speaker recognition [14] by building a noisy model and using it to decode noisy test segments. The use of PMC inside modern speaker recognition i-vector systems is complex, as the noise has to be injected inside all the different models: UBM, i-vector extractor and scoring models. But in practice, the high computational expense, mainly in the scoring model, of such a procedure makes it unfeasible in practice. A robust backend training method called "multi-style" [15] was proposed as a possible solution to account for the noise in the scoring phase. This method uses a large set of clean an noisy data affected with different noises and SNR levels to build a generic scoring model. The model obtained yields good performance in general, but is still suboptimal for a particular noise because of its generalization (the same system is used for all noises). Another problem with this approach is that it also assumes (theoretically) that test noise is in some way present in the training data, which is not always true. Finally, the use of deep neural networks (DNNs) has been investigated for robust speaker recognition before being successfully applied to speech recognition [16, 17, 18, 19]. DNNs have been used either to improve the speaker model (like the "d-vectors" model proposed in [20] and extracted from the last hidden layer of a DNN) or to improve the computation of the i-vectors statistics in noisy conditions [21]. But in spite of the extensive training time needed to build such models, no significant improvements were observed compared to the previously cited methods.

This paper is an extension of our work in [22] where we proposed an i-vector "denoising" technique, we called i-MAP, in order to deal with additive noise. The advantage of this approach is that we can use a regular clean backend since the resultant i-vectors are assumed to be noise free. In order to build this system, a number of assumptions are made over the clean i-vectors and the noise distribution in the i-vector space. We assume that both clean i-vectors and noise are normally distributed in the i-vector space. The first hypothesis is justified by the factor analysis model used to extract the i-vectors [23] which assumes a normal distribution for the resulting i-vectors. Regarding the noise, Gaussian distribution modeling seems to be suitable. Even though, the noise is theoretically known to be non-additive in the i-vector space, an additive noise model seems to give encouraging results. It shows an improvement by up to 60% in the recognition performance compared to the baseline system and by nearly 30% compared to the "multi-style" scoring regime. In addition, the approach is extensible to a mixture of Gaussians to model the noise in i-vector space. The originality of this technique is that it not only uses information about the noise but also information about clean i-vectors (the corresponding probability density functions in the i-vector space). Indeed, compared to different approaches such as MMSE estimators, this technique does not just model the relationship between clean and noisy i-vectors (through the noise model), but also gives information about the clean i-vectors distribution. Hence, it incorporates rich information the clean i-vectors properties and minimizes the risk of producing distorted estimates.

In this paper, we introduce an implementation of the proposed method in an i-vector-based speaker recognition system. Then, we propose an extension based on building a noise distribution database in the i-vector space in an off-line step to reduce the computational time imposed by the i-MAP denoising technique. This way, the test noise distribution parameters in the i-vector space are approximated by one of the available distributions in the database instead of being directly computed using the noise frames present in the test segment. A distribution selection scheme is described based on distance measure between

a given noisy test i-vector and all noisy i-vectors distributions present in the database. We show that we still achieve almost the same results while making the i-vector cleaning much faster. Furthermore, i-MAP does not deteriorate the resultant error rate when applied on clean speech, which constitutes a very interesting robustness aspect of the technique proposed.

This paper is structured as follows. Section 2 describes the i-vector framework for speaker recognition. Section 3 details the proposed i-MAP denoising approach. Section 4 presents the experimental protocol. Section 5 details the estimation method of the system parameters. Section 6 details the integration procedure of the method proposed in a speaker recognition system. Section 7 details the structure of the VAD system used in our work. Section 8 presents the system performance after the use of the i-MAP compensation. Finally, Section 9 proposes a technique to speed-up the noise distribution estimation in the i-vector space and Section 10 presents the corresponding results.

2. The i-vector framework

In this section we present the i-vector framework along with the scoring procedure that will be used further in our experiments.

2.1. The total-variability subspace

The i-vector paradigm was motivated by the existing super-vector-based joint factor analysis (JFA) approach [24, 25]. While the JFA approach models the speaker and channel variability space separately, i-vectors are formed by modeling a single low-dimensional total-variability space that covers both the speaker and channel variability [23]. In this approach, an i-vector extractor converts a sequence of acoustic vectors into a single low-dimensional vector representing the whole speech utterance. The speaker- and session-dependent super-vector s of concatenated Gaussian Mixture Model (GMM) means is assumed to obey a linear model of the form:

$$s = m + Tw \tag{1}$$

where :

- m is the mean super-vector of the Universal Background Model (UBM)
- T is the low-rank variability matrix obtained from a large dataset by MAP estimation [24]. It represents the total variability subspace.
- w is a normally distributed latent variable called "i-vector".

Extracting an i-vector from the total variability subspace is essentially a maximum a-posteriori adaptation of w in the space defined by T . The algorithms for the estimation of T and the extraction of i-vectors are described in [26].

2.2. The i-vector scoring system

Many dimensionality reduction techniques (such as LDA) and generative models (like PLDA, and the Two-covariance model) have been developed in order to improve the i-vector comparison in speaker verification trials. The speaker verification score given two i-vectors w_1 and w_2 is the likelihood ratio described by:

$$score = log \frac{P(w_1, w_2|\theta_{tar})}{P(w_1, w_2|\theta_{non})} \tag{2}$$

where the hypothesis θ_{tar} states that inputs w_1 and w_2 are from the same speaker and the hypothesis θ_{non} states they are from different speakers.

Below, we focus on the generative model that we used in our work: the two-covariance scoring model.

2.2.1. The two-covariance scoring model

This model is a particular case of the Probabilistic Linear Discriminant Analysis (PLDA) described in [27]. It can be seen as a scoring method and a convolutive noise compensation technique. It consists of a simple linear-Gaussian generative model in which an i-vector w of a speaker s can be decomposed in:

$$w = y_s + \varepsilon \tag{3}$$

where the speaker model y_s is a vector of the same dimensionality as an i-vector, ε is Gaussian noise and :

$$P(y_s) = \mathcal{N}(\mu, B) \qquad (4) \qquad\qquad P(w|y_s) = \mathcal{N}(y_s, W) \qquad (5)$$

$\mathcal{N}$ denotes the normal distribution, μ represents the overall mean of the training data set, B and W are the between- and within-speaker covariance matrices defined as :

$$B = \sum_{s=1}^{S} \frac{n_s}{n}(y_s - \mu)(y_s - \mu)^t \qquad (6)$$

$$W = \frac{1}{n}\sum_{s=1}^{S}\sum_{i=1}^{n_s}(w_i^s - y_s)(w_i^s - y_s)^t \qquad (7)$$

where n_s is the number of utterances for speaker s, n is the total number of utterances, w_i are the i-vectors of sessions of speaker s, y_s is the mean of all the i-vectors of speaker s and μ represents the overall mean of the training data set. Under assumptions (6) and (7), the score from Equation (2) can be expressed as:

$$score = \frac{\int \mathcal{N}(w_1|y, W)\mathcal{N}(w_2|y, W)\mathcal{N}(y|\mu, B)dy}{\prod_{i=1,2}\int \mathcal{N}(w_i|y, W)\mathcal{N}(y|\mu, B)dy} \qquad (8)$$

the explicit solution of (8) is given in [28].

3. I-vector denoising using MAP

This section is dedicated to the description of our new i-vector "cleaning" technique, "i-MAP". De-noising the i-vector directly allows to use classical state-of-the-art scoring models based on generative models like two-covariance [28], Gaussian-PLDA [27] or heavy tailed PLDA [29] estimated using clean data without any adaptation to the test noise. This also makes our method equally valid in matched and mismatched conditions between enrollment and test since it "cleans" a noisy i-vector without introducing any distortions on a clean one.

Formally, given a noisy i-vector Y_0, our goal is to estimate the corresponding clean version X_0. Let's define two random variables X and Y corresponding respectively to the clean and noisy i-vectors. We define the noise random variable N by:

$$N = Y - X \tag{9}$$

We consider that clean i-vectors X are normally distributed as described in [23], and assume that noise (N) can also be represented by a normal distribution in the i-vector space. We can then define the corresponding probability distribution functions $f(X)$ and $f(N)$ as :

$$f(X) = \mathcal{N}(\mu_X, \Sigma_X) \tag{10}$$

$$f(N) = \mathcal{N}(\mu_N, \Sigma_N) \tag{11}$$

where $\mathcal{N}(\mu, \Sigma)$ denotes a normal distribution with mean μ and full covariance matrix Σ.

Referring to (9),(10) and (11) we can express $f(Y_0|X)$ for a given Y_0 as:

$$f(Y_0|X) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma_N|^{\frac{1}{2}}} exp^{-\frac{1}{2}(Y_0 - X - \mu_N)^t \Sigma_N^{-1}(Y_0 - X - \mu_N)} \tag{12}$$

Based on the noise model (9) and the two previously defined distribution, we can estimate, for a given noisy i-vector Y_0, its clean version $\hat{X}_0$ using a MAP estimator :

$$\hat{X}_0 = \underset{X}{\text{argmax}}\{\ln f(X|Y_0)\} \tag{13}$$

Using the Bayesian rule, we can write $f(X|Y_0)$ as :

$$f(X|Y_0) = \frac{f(Y_0|X)f(X)}{f(Y_0)} \tag{14}$$

After combining (13) and (14) :

$$\hat{X}_0 = \underset{X}{\text{argmax}}\{\ln f(Y_0|X)f(X)\} \tag{15}$$

Finding $\hat{X}_0$ becomes equivalent to solving:

$$\frac{\partial}{\partial X}\{\ln f(Y_0|X) + \ln f(X)\} = 0 \tag{16}$$

By developing (16) using (12), we end up with:

$$\frac{\partial}{\partial X}\{(Y_0-X-\mu_N)^t\Sigma_N^{-1}(Y_0-X-\mu_N)\}+\frac{\partial}{\partial X}\{(X-\mu_X)^t\Sigma_X^{-1}(X-\mu_X)\}=0 \quad (17)$$

After the derivation, the final expression of the clean i-vector $\hat{X}_0$, given the noisy version Y_0 and both X and N distribution parameters, is:

$$\hat{X}_0=(\Sigma_N^{-1}+\Sigma_X^{-1})^{-1}(\Sigma_N^{-1}(Y_0-\mu_N)+\Sigma_X^{-1}\mu_X) \quad (18)$$

In i-vector -based speaker recognition systems [23], length normalization was shown to improve the overall performance [30]. In our case, it is important to mention that all the noisy and clean i-vectors used were initially length-normalized.

4. Experimental protocol

Our experiments operate on 19 Mel-Frequency Cepstral Coefficients (plus energy) augmented with 19 first (Δ) and 11 second ($\Delta\Delta$) derivatives. A mean and variance normalization (MVN) technique is applied on the MFCC features estimated using the speech portion of the audio file. The low-energy frames (corresponding mainly to silence) are removed. The decision boundary used to determine speech frames is computed for each utterance as explained in Section 7 and the value of α used in all experiments for voice activity detection is $\alpha = 0$ and for noise extraction $\alpha = 1$ (explained in Section 7).

Two SR systems are used in our experiments depending of the speakers gender in enrollment/test data. Two gender-dependent 512 diagonal component UBMs and total variability matrices of low rank 400 are estimated using NIST SRE 2004, 2005, 2006 and Switchboard data. The male models (male UBM and total variability matrix) were trained using 15660 utterances corresponding to 1147 speakers and the female models (female UBM and total variability matrix) were trained using 24100 utterances corresponding to 2012 speakers. The LIA_SpkDet package of the LIA_RAL/ALIZE toolkit is used for the estimation of the total variability matrix and the i-vector extraction. The algorithms used

are described in [26]. Finally a two-covariance-based scoring [28] is applied. For each gender, the equal-error rate (EER) over the NIST SRE 2008 test data on the "short2/short3" task under the "det7" conditions [31] will be used as a reference to monitor the performance improvement of two systems in noisy conditions compared to the baseline system : noisy PLDA backend (PLDA model trained using data affected with the noise type and SNR level present in test/enrollment data) and the "multi-style" backend. For the noisy PLDA system, the eigenvoice rank used is equal to 100 and the eigenchannel matrix is kept full-rank (400). PLDA is preceded by 2 iterations of LW-normalization (spherical nuisance normalization [32]).

We use 18 noise samples from the free sound repository FreeSound.org [33] as background noises (used to alter test/enrollment data and build the noise distribution). The open-source toolkit FaNT [34] was used to add these noises to the full waveforms generating new noisy audio files for each noise / SNR level.

5. Estimation of $f(X)$ and $f(N)$

In this section, we will work with a total of six configurations: two different noises (crowd and air-cooling) and three SNR levels (10dB, 5dB and 0dB) using 3000 clean train speech segments ($SNR > 25dB$).

The clean i-vectors distribution $f(X)$ and the noise distribution $f(N)$ are the two most important components in this denoising procedure. $f(X)$ has the advantage of being noise-independent, so it could be estimated once and for all over a large set of clean i-vectors in an off-line step initially before performing any compensation.

On the other hand, $f(N)$ makes the system able to adapt to the noise present in the signal and compensate its effect more effectively. It is estimated for each different test noise and it requires the existence of clean i-vectors and the noisy versions corresponding to the same segments. First, for the clean part and once the train files are fixed, the corresponding clean i-vectors (X) are extracted. Then, for a given noisy test segment, the noise is extracted from the signal

(using a VAD and selecting the low-energy frames) then added to the clean train audio files. Finally, the corresponding noisy i-vectors (Y) are estimated and Equation (9) is used to compute N then $f(N)$.

Below, we focus on minimizing the number of train files used to build $f(N)$ along with their selection criteria.

5.1. Number of i-vectors needed to estimate $f(N)$

In a "clean enrollment / noisy test" setup and for each of the six previously-described noise configurations, the EER is evaluated for male data using a different number of train i-vectors to estimate $f(N)$ going from 400 to 3000. Each time, $N = Y - X$ is used prior to the scoring phase using the selected i-vectors to estimate μ_N and Σ_N. For each length, Figure 1 shows the EER obtained on 10 different lists picked randomly from the train i-vectors set :

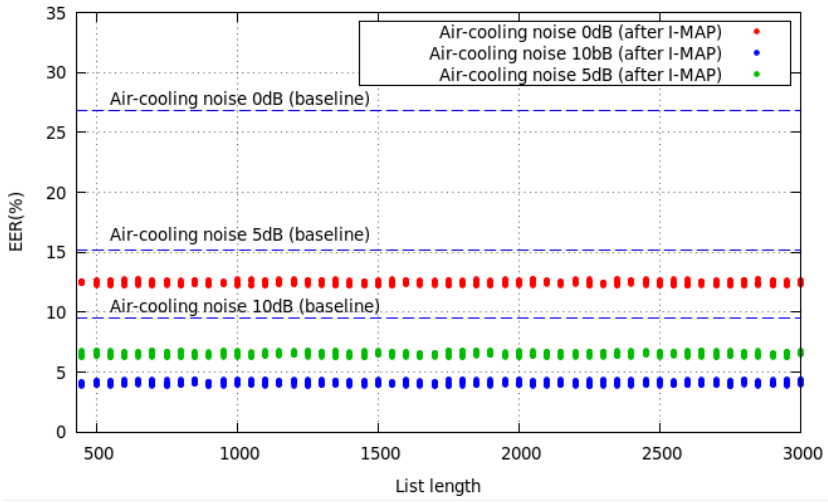

Figure 1: EER variation with the amount of i-vectors used to estimate the noise distribution $f(N)$ for the "air-cooling noise" at 0dB, 5dB and 10dB for male data (10 measures for each length).

The x-axis in Figure 1 starts at 450 since the i-vector space dimension used in this experiment is 400. Indeed, going below 400 might cause singular covariance matrices (used in Equation 18).

It is clear that for the three SNR levels, the EER does not vary much beyond 500. Therefore, we will set the noise model training set size to 500 i-vectors for our next experiments.

5.2. Train i-vector selection for the noise density estimation

The goal of this experiment is to find a criterion that improves "globally" the quality of the cleaned-up i-vectors without putting strict constraints on the test segments duration or content. In this subsection, only male telephone recordings are used for both train and test for simplification purposes. Similar findings have been observed for female data and the same criterion will be used for the two genders.

Once set to 500 the number of i-vectors needed to estimate $f(N)$, we concentrate on their selection criteria. For the six different configurations, we created a set of 300 lists of 500 elements picked randomly from the original set of 3000 clean audio files which will be used to estimate $f(N)$. For each list, we plot the resultant EER after compensation according to the average files duration. Figure 2 shows the curve obtained using noisy test data affected with crowd-noise on 10dB.

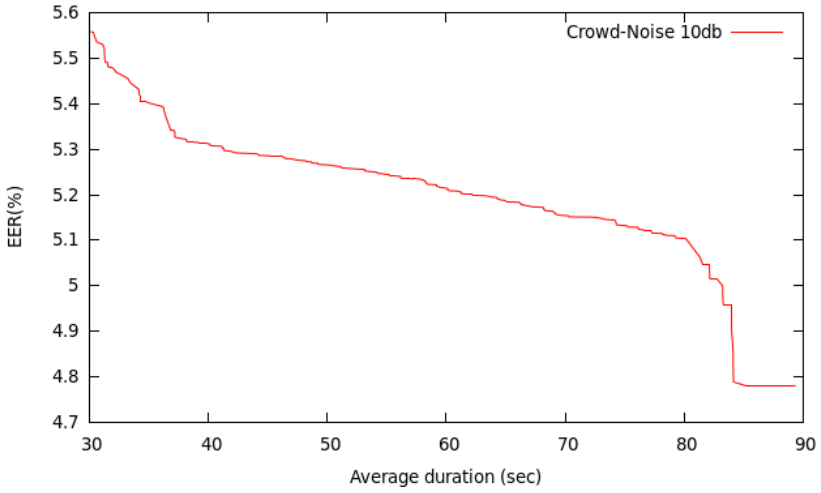

Figure 2: EER variation with the average speech duration of the segments used to estimate $f(N)$ for the "crowd-noise" on 10dB.

It is easy to see that the longer speech segments produce better results than the shorter ones. In the rest of this paper, the longest 500 files will be used as a train set to estimate $f(N)$. The sharp fall in Figure 2 is a result of the train data available (few long train recordings).

5.3. Compensation threshold

One of the biggest advantages of the i-MAP compensation scheme is that it does not affect clean i-vectors or deteriorate the associated error-rates. Therefore, in order to save time and avoid unnecessary compensations, we can fix an SNR threshold beyond which no transformation is applied.

In order to set the value of the SNR threshold beyond which a test utterance is considered clean, we study the variation of the EER with the maximum denoised test segment SNR. Figure 3 shows that attempting to denoise i-vectors corresponding to noisy segments having an SNR greater than 25dB will not improve the end result much. The variation of the equal error rate obtained after compensation with the SNR threshold is given for two different noises.

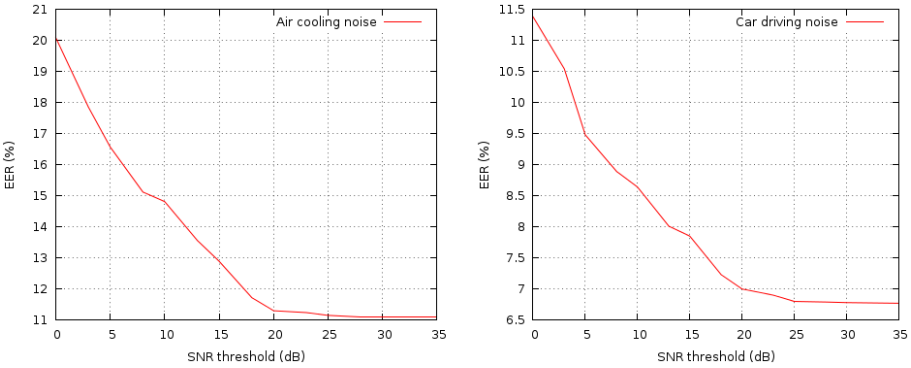

Figure 3: Variation of the EER (after i-MAP compensation) with the SNR threshold in dB for two different noises (car driving noise and air-cooling noise) using clean enrollment data and noisy test affected with the same noise and different SNR levels from 0dB to 35dB.

In the next sections, we will use $SNR_{threshold} = 25dB$ to decide whether the denoising procedure is required or not.

6. Integration of the denoising method in a speaker recognition system

The new i-vector denoising method allows to build a speaker recognition system that takes into account the test signal SNR level as shown in Figure 4.

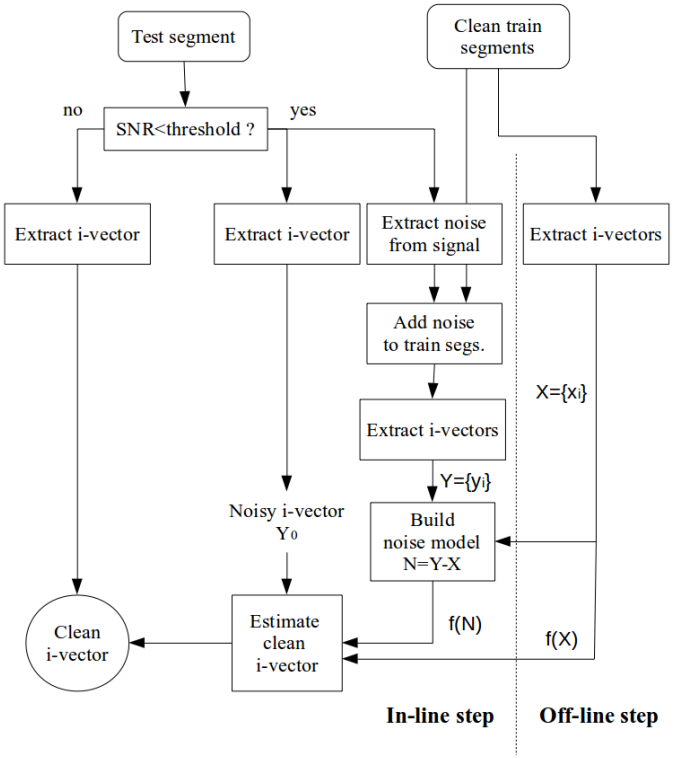

Figure 4: Clean i-vector extraction algorithm. Firstly, the signal SNR level is estimated. Then, if the segment is considered noisy ($SNR < threashold$), the corresponding noise distribution is estimated in the i-vector space. Finally, the i-MAP denoising procedure is applied.

Before starting, an SNR threshold above which a segment is considered clean has to be set (in our experiments, we used $SNR_{threshold} = 25dB$).

Then, the algorithm follows these steps:

- **SNR checking:** The SNR level is estimated for the test segment and compared to the threshold.

- **The clean case:** If the segment is clean, then a standard i-vector extraction is done.
- **The noisy case:** If the segment is noisy:
    1. The corresponding noisy i-vector Y_0 is computed.
    2. A VAD is used to extract the noise part from the signal by selecting the low-energy frames in the signal corresponding to the non-speech intervals. The structure of the VAD system used in our experiments is detailed in Section 7 along with the decision threshold for voice activity detection and noise extraction.
    3. The noise is added to the set of clean train files in the time domain with the SNR of the test utterance (estimated in the first step).
    4. A standard i-vector extraction is done using the noisy train files (corresponding to the Y data).
    5. The noise distribution $f(N)$ in the i-vector space is estimated using Equation (9).
    6. The new clean i-vector is estimated using Equation (18).

It is important to mention that in noisy environments with low SNR levels, the voice activity detection procedure becomes less accurate which might affect the quality of the noise estimate (noise extracted from the signal). In our experiments, using two different thresholds (one for voice activity detection and another for noise extraction) has been used as a partial solution to deal with this problem. For each task, we try to select the most useful frames (based on their energy), hence reducing the risk of recognition error. The next section describes the VAD system used in this paper.

7. VAD system configuration for speech and noise detection

The VAD system is a central component in our technique as explained in Section 6 and its performance influences greatly the efficiency of the algorithm described in Figure 4. It is used to select the most useful speech frames (prior

to the i-vectors estimation process) and determine noise segments (in the noise extraction process (Figure 4)). In the next subsection, we detail the structure of the VAD system used in our work, the decision process (speech/non-speech) and explain its use to extract noise from noisy segments.

7.1. Voice activity detection

The most commonly used VAD systems in speaker recognition are energy-based and that is due to their efficiency and fast computation compared to other techniques [35, 36]. The highest energy frames are usually used for speaker verification since they contain the most useful information and are more likely to resist environment disturbances.

The VAD system used in this paper is described in [36, 37]. It is based on the log-energy distribution of frames. First, the log-energies of each frames of an utterance are computed. Then, using the EM algorithm, the distribution of log-energy coefficients is estimated using a Gaussian mixture model with 3 components. Frames which correspond to the highest mean Gaussian (high energy frames) are then used as speech frames while low energy frames, corresponding mainly to silence and noise, are discarded. A threshold is computed to determine the decision making boundary between speech and non-speech class as defined in Equation (19):

$$\tau_{thr} = \mu_i - \alpha \ \sigma_i \tag{19}$$

Where μ_i and σ_i are the mean and standard deviation of the Gaussian corresponding to high-energy frames, and α is a value controlling the selectivity. Increasing the value of the coefficient α allows to take more high-energy frames into account. Finally, the selection of frames is then smoothed using a morphological window [37]. Figure 5 shows an example of a 3-components GMM approximation of log-energy distribution and the threshold used to select speech/non-speech frames.

In all our experiments, we used $\alpha = 0$ during the voice activity detection process in order to have a strict selection of speech frames and minimize the

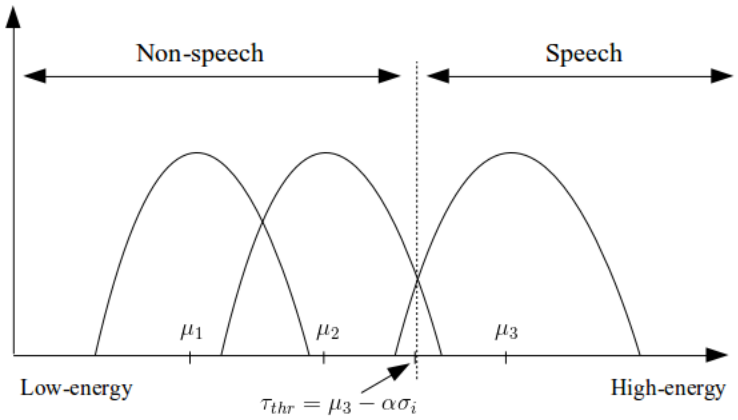

Figure 5: Log-energy distribution and speech threshold.

risk of selection error between speech and non-speech frames that might occur in low SNR conditions (eg. 0dB).

7.2. Noise extraction

The VAD system developed can be used to extract an estimate of the background noise from noisy segments. Indeed, low energy frames (corresponding mainly to silence) can be extracted by taking the complement of speech segments.

This procedure is sufficient for segments with average and high SNR levels ($>$ 10dB), but in low SNR levels (eg. 0dB), it becomes hard to decide with certainty whether a frame corresponds to speech or noise. For this reason, a more strict configuration is used when we want to extract noise from noisy signals. In all our experiments, this procedure is used to extract noise from a signal :

1. Compute the log-energy value of all frames.
2. Model the log-energy distribution using a 3-components GMM using the EM algorithm.
3. Compute τ_{thr} using Equation 19 with $\alpha = 1$.
4. Take all frames that correspond to $log\text{-}energy < \tau_{thr}$ as noise frames.

Setting $\alpha = 1$ while extracting noise allows to consider more high-energy frames as speech, so less low-energy frames as noise, hence minimizing the risk of selecting speech frames as silence in low SNR conditions. It is important to mention that this procedure is used as a partial solution and that it does not prevent the system from producing poor noise estimates in low SNR conditions.

8. Recognition performance using i-MAP

In this section, the new estimated clean i-vectors (corresponding to either test or enrollment segments) will be referred to as "I-MAP" vectors. The LIA speaker verification baseline system reaches an EER=1.59% for male test and EER=2.66% for female test in clean conditions.

The enrollment and test data have been altered using two different sets of noises {nature noise, rain and engine noise} for enrollment and {air-cooling, car-driving and crowd-noise} for test at 4 different SNR levels: 0dB, 5dB, 10dB and 15dB.

We will compare 4 systems performances in this section. It is important to remind that two versions were built for each system depending on the gender of speakers present in test/enrollment. The training data described in the experimental protocol have been used to train each model independently (1147 speakers for the male models and 2012 speakers for the female models).

- Noisy i-vectors used with the baseline system (clean backend ; one for each gender).
- Noisy i-vectors used with a multi-style backend : The multi-style backend is trained using 5 different noises {applause, ringing, bus station background noise, ocean wave noise and rainstorm noise} (different from the ones used to affect enrollment and test data) at different SNR levels picked randomly from 0dB to 25dB (to each noisy train i-vector corresponds one noise and a fixed SNR level). This setup can be understood as a "partial multi-style training" since only the scoring model is affected by noise (the world model and i-vector extractor are trained using clean data).

- Noisy i-vectors used with a noisy PLDA : In this system, two gender-dependent PLDA models have been trained with data affected with the noise type and SNR level present in test/enrollment data (one for male used on male test/enrollment data and another for female used on female test/enrollment data). The eigenvoice rank used in these models is equal to 100 and the eigenchannel matrix is kept full-rank (400). PLDA is preceded by 2 iterations of LW-normalization (spherical nuisance normalization [32]).

- I-MAP vectors used with a clean backend (the algorithm described in Section 6 is used for each i-vector and the noisy i-vectors corresponding to male and female speakers are treated separately).

We present first the system performance using clean enrollment data, then we compare them with the results given in different noisy enrollment configurations.

8.1. System performance using clean enrollment data and noisy test data:

For three different test noises, Tables 1 and 2 show respectively the male and female systems performance when used on clean enrollment and noisy test data. In order to evaluate the impact of the voice activity detection procedure on the performance of these 4 systems, we add a "oracle VAD" condition in which the "true" VAD labels obtained using clean speech are used to compute all noisy i-vectors.

When i-MAP compensation is used, a relative improvement range between 48% and 64% is observed for both genders, whereas the "multi-style" compensation is limited to 28% as a maximum relative improvement compared to the baseline system. The noisy PLDA system outperforms the "multi-style" backend (reaching 33% or relative improvement) but is still not as good as i-MAP. Training such a system (noisy PLDA) requires a large number of training sessions which is not suitable for real applications. On the other hand, fewer training sessions are used in i-MAP while giving better results. This experiment proves clearly our method's potential in mismatched conditions.

Table 1: Recognition performance in different test conditions using clean enrollment and noisy test for male data. In the "oracle VAD" condition, the true VAD labels (obtained using clean speech) are used to compute noisy i-vectors whereas in the "real VAD" condition, the real labels (obtained using noisy speech) are used.

Test condition		**EER(%)**							
		Baseline		**Multi-style**		**Noisy PLDA**		**i-MAP**	
		Real VAD	*Oracle VAD*	*Real VAD*	*Oracle VAD*	*Real VAD*	*Oracle VAD*	*Real VAD*	*Oracle VAD*
Air-cooling noise	**0dB**	26.85	21.44	23.53	19.46	22.01	17.95	13.21	11.34
	5dB	15.21	12.10	12.21	9.97	12.92	11.23	7.25	6.02
	10dB	9.51	7.40	8.62	7.00	7.32	6.11	4.85	4.12
	15dB	5.41	4.24	4.72	3.84	4.65	3.77	2.85	2.51
Car driving noise	**0dB**	25.54	20.01	22.85	18.98	22.21	19.14	12.05	10.11
	5dB	14.54	11.55	10.54	8.94	11.63	9.58	6.65	5.86
	10dB	8.32	6.65	7.24	6.06	6.40	5.30	3.78	3.21
	15dB	4.82	3.72	4.20	3.52	4.14	3.47	2.36	2.25
Crowd-noise	**0dB**	24.24	19.29	22.03	17.91	20.60	17.47	11.55	11.01
	5dB	13.94	10.94	10.01	8.39	10.73	8.83	5.09	4.50
	10dB	7.77	6.19	5.97	4.97	6.75	5.80	3.05	2.57
	15dB	4.01	3.15	3.82	3.22	3.12	2.58	2.02	1.71

While comparing the "Real VAD" and the "Oracle VAD" conditions in Tables 1 and 2, a relative EER loss ranging between 15% and 22% is observed for low SNR levels (0dB and 5dB) and a relative loss ranging between 10% and 20% is observed for higher SNR levels (10dB and 15dB). These results illustrate the impact of our VAD system especially in low SNR levels where it becomes more difficult to distinguish speech from noise frames.

8.2. System performance using noisy enrollment data and noisy test data:

In this subsection, we present the system performance when used on noisy data in enrollment and test. Figures 6 and 7 give the performance of the four

Table 2: Recognition performance in different test conditions using clean enrollment and noisy test data for female data. In the "oracle VAD" condition, the true VAD labels (obtained using clean speech) are used to compute noisy i-vectors whereas in the "real VAD" condition, the real labels (obtained using noisy speech) are used.

Test condition		**EER(%)**							
		Baseline		**Multi-style**		**Noisy PLDA**		**i-MAP**	
		Real VAD	*Oracle VAD*	*Real VAD*	*Oracle VAD*	*Real VAD*	*Oracle VAD*	*Real VAD*	*Oracle VAD*
Air-cooling noise	**0dB**	27.19	21.64	22.56	18.46	21.10	18.23	11.69	10.39
	5dB	16.77	13.32	14.92	12.58	13.68	11.13	7.37	6.21
	10dB	9.01	7.15	6.93	5.78	8.19	6.70	4.05	3.36
	15dB	6.42	4.96	5.58	4.56	4.18	3.45	2.82	2.39
Car driving noise	**0dB**	24.82	19.21	19.85	16.35	18.26	15.81	9.43	7.88
	5dB	14.90	11.79	12.06	9.79	10.55	8.96	5.66	5.25
	10dB	8.65	6.70	6.83	5.55	8.35	6.68	3.25	2.77
	15dB	5.89	4.56	4.71	3.85	3.14	2.64	2.41	2.14
Crowd-noise	**0dB**	25.44	19.63	20.86	17.39	19.54	16.69	11.44	9.86
	5dB	14.37	11.31	12.07	9.90	13.65	10.99	6.32	5.57
	10dB	8.77	6.85	7.10	5.99	5.60	4.858	3.68	3.26
	15dB	5.78	4.46	4.68	3.84	3.37	2.79	2.60	2.25

systems in different SNR scenarios for male and female data respectively. The noise is picked randomly from {nature noise, rain and engine noise} for enrollment and {air-cooling, car-driving and crowd-noise} for test. In this setup, the noisy PLDA backend largely outperforms the "multi-style" system since it is more adapted to the enrollment/test noise. Such system would need prior knowledge about test noise and a large training set altered using the same noise. Such constraints are hard to achieve in real life applications. Also, it is clear that i-MAP outperforms by far the "multi-style" scoring method and performs better than noisy PLDA in all conditions. Indeed, an average relative improvement of 43% is observed in all conditions compared to the baseline performance

and of 28% compared to a "multi-style" backend performance. It is important to see that while certain noise compensation techniques lose their efficiency in low-SNR conditions (such as the RATZ [11] algorithm), the i-MAP compensation scheme still reaches important gain in low SNR levels (near $0dB$). In conclusion, the results show clearly the potential of the method proposed in various conditions while using different noises.

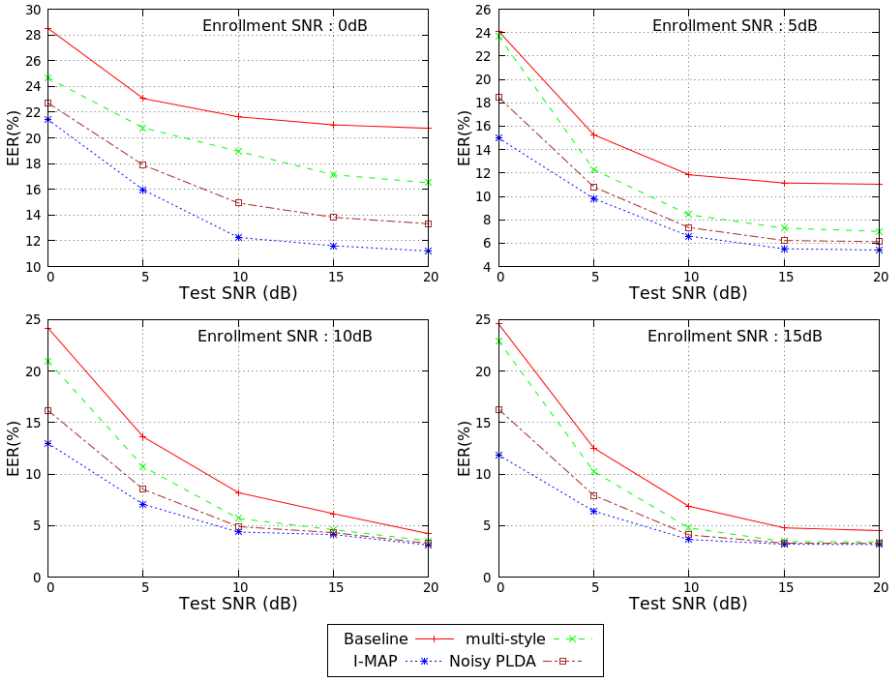

Figure 6: Performance on male data : Each figure corresponds to a different enrollment SNR. The x-axis corresponds to the SNR level in the test segments and the y-axis gives the resultant EER.

8.3. System performance in a heterogeneous setup:

We performed another experiment to prove the validity of our technique in a situation where the noise level is varying randomly between the enrollment/test segments. In this experiment, all the speech files (for enrollment and test) are corrupted by a noise with a varying randomly-selected SNR level between 0dB to 20dB. Table 3 shows the obtained results with the three systems.

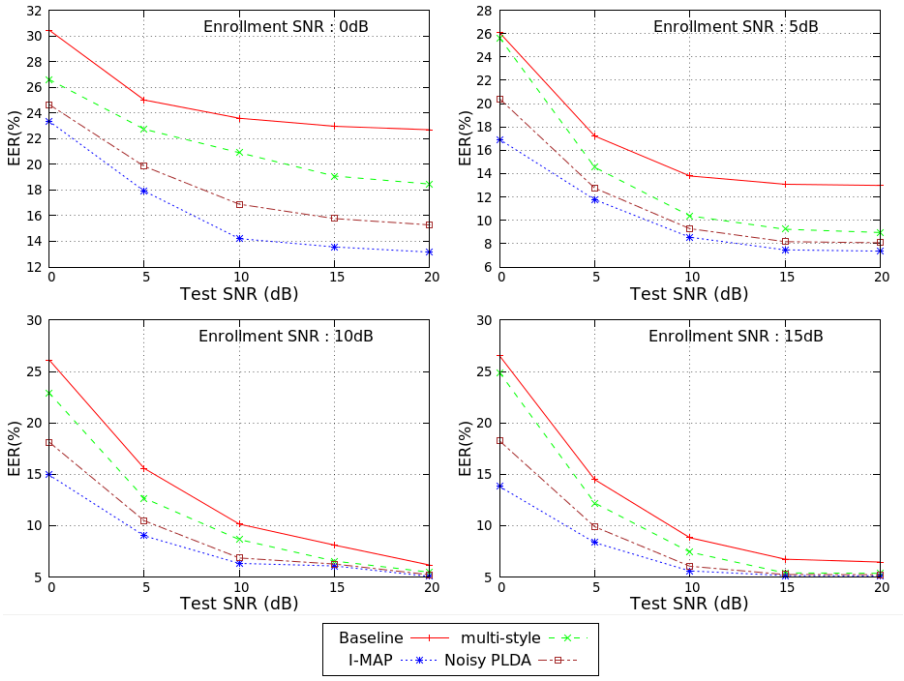

Figure 7: Performance on female data : Each figure corresponds to a different enrollment SNR. The x-axis corresponds to the SNR level in the test segments and the y-axis gives the resultant EER.

Due to the large variability in terms of noise and SNR level, a significant improvement is observed in this condition using i-MAP with a clean backend compared to the "multi-style" scoring used with noisy i-vectors. In fact, this shows the limits of the "multi-style" scoring model due to its generalization property. This makes our method more efficient in unknown test/enrollment conditions since it adapts itself to any given noise and level present in a test segment. The noisy PLDA backend outperforms the "multi-style" system but cannot be used in real applications since it assumes prior knowledge about test/enrollment conditions and requires adding noise to a large set of training sessions. The difference between the two systems (multi-style and noisy PLDA) in this experiment is that the former is built using clean and noisy segments affected by noises which does not appear in test/enrollment conditions while the

Table 3: Performance comparison in a heterogeneous setup.

	EER (%)	
	Male	**Female**
Baseline	29.65	28.82
"multi-style" backend	23.12	22.86
Noisy PLDA backend	20.72	20.96
I-MAP + clean backend	16.27	15.76

latter is built using test/enrollment noises at different SNR levels. This explains the difference between their performance.

9. Noisy i-vector distribution database for i-vector denoising

In this section, we introduce the use of a noise distribution database in the i-vector space to speed-up the denoinsing process. We present the new system's layout along with its configuration.

9.1. Motivation

In real world applications, computational time and memory requirements are two important factors to consider, especially for critical applications such as forensics and light-memory devices such as smartphones. For a test utterance containing a certain noise N_k, using the method proposed in this paper to estimate the noise distribution hyperparameters $d_{N_k} : (\mu_{N_k}, \Sigma_{N_k})$ is time-consuming and computationally expensive due to the number of steps required (adding noise to the train files, noisy i-vectors extraction then estimation of the noise distribution in the i-vector space).

To deal with this problem, we propose a solution that avoids the in-line noise distribution estimation step by using a noise distribution database in the i-vector space built off-line prior to the recognition phase. Instead of estimating the noise distribution directly from the noisy test signal (extracting the noise frames then using them to build a noisy i-vector Gaussian distribution affected

by the same noise), we try to find the best approximation of its distribution among the ones present in our database. For a given noisy test i-vector Y_0, we usually do not have the corresponding clean version X_0. So, we cannot base our distribution selection process on the corresponding noise ($N_0 = Y_0 - X_0$). A possible solution to this problem is to store, for each configuration present in the database, both the noisy i-vector distribution d_{Y_k} (which will be used for the distribution selection) and the noise distribution d_{N_k} (which will be used for the i-MAP compensation). For a given noisy test i-vector Y_0, the most likely noisy i-vector distribution $d_{Y_k} : (\mu_{Y_k}, \Sigma_{Y_k})$ is first selected from the database. Then, the corresponding noise distribution $d_{N_k} : (\mu_{N_k}, \Sigma_{N_k})$ in the i-vector space is used for the denoising as shown in Figure 8.

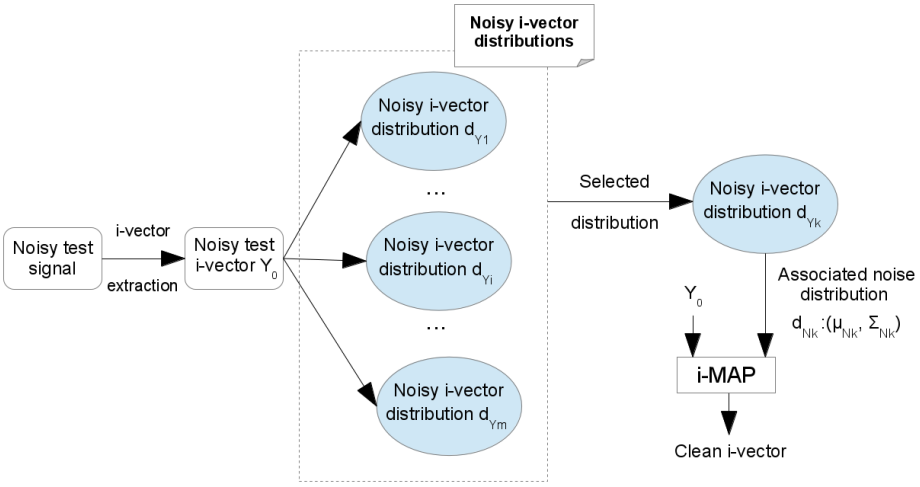

Figure 8: Using a noise distribution database in the i-vector space for the denoising. First, the noisy test i-vector is extracted. Then, the most likely noisy i-vector distribution d_{Y_k} is selected. Finally, the corresponding noise distribution d_{N_k} is used to perform an i-MAP compensation.

In the next subsection, we present the method used to build the database, the noise distribution selection criteria and the resulting new compensation scheme. Only male data (in test, enrollment and training) will be used in our analysis for clarity and simplification purposes. The same results and techniques can be

transposed to female data in real applications.

9.2. Building the database

The noise database is built using 18 different noises coming from different environments (wind, music, car driving noise, engine noise, applause, air cooling noise, crowd noise, ..) and 13 SNR levels variating from 0dB to 30dB. It is important to mention that noises used to build the database are completely different from the ones used to alter enrollment/test data. This condition allows us to simulate a real world scenario. For each different noise and SNR level, we follow the steps described in Algorithm 1. We end up with 234 different Gaussian distribution of noisy i-vectors. The next step is to select the most likely distribution for a given noisy test i-vector Y_0.

Algorithm 1 Building the noise distribution database in the i-vector space.

for each $(noise_i, SNR_j)$ **do**

1 - Add $noise_i$ at the level SNR_j to the clean train files.

2 - Extract the noisy i-vectors Y_{ij} corresponding to the noisy segments.

3 - Compute the associated noise data in the i-vector space : $N_{ij} = Y_{ij} - X$

4 - Compute the noise distribution hyperparameters $d_{N_{ij}} : (\mu_{ij}, \Sigma_{ij})$.

end for

As mentioned before, the noisy i-vector distribution hyperparameters $d_{Y_{ij}} : (\mu_{ij}, \Sigma_{ij})$ are also stored since they will be needed in the noise distribution selection step.

9.3. Noise distribution selection

Selecting the correct noisy i-vector distribution is crucial in order to have the best possible results. We consider that each condition present in the database $(noise_i, SNR_j)$ corresponds to a different noise and try to select the closest one to a given noisy test i-vector Y_0 based on a distance measure. A natural choice for similarity measure in this context is the likelihood since we supposed

that the noisy i-vector distribution is Gaussian. Another possible choice is the n-dimensional Euclidean distance between a noisy i-vector and the mean of a noisy i-vectors distribution. This distance has the advantage of being extremely fast compared to the likelihood measure and could be a better choice in real-time applications since it requires much less computation.

The selected distribution d_p used to denoise a noisy i-vector Y_0 would be:

$$d_p = \underset{d_i}{\operatorname{argmin}} \{dist(Y_0, d_i)/i \in \{1, .., nb\_distribution\}\} \tag{20}$$

- **Euclidean distance:** The Euclidean distance could be used between a noisy i-vector and the means of all noisy distributions. For a given noisy test i-vector Y_0, and a noisy i-vector distribution k : $d_k \sim \mathcal{N}(\mu_{Y_k}, \Sigma_{Y_k})$, the distance to be used is :

$$dist_{Eucl}(Y_0, \mu_{Y_k}) = \left(\sum_{i=1}^{n} (Y_{0i} - \mu_{Y_{ki}})^2 \right)^{\frac{1}{2}} \tag{21}$$

- **Likelihood measure:** Using the likelihood of a noisy i-vector Y_0 with respect to all noisy i-vectors distributions d_k is a natural choice. This measure accounts for the noisy i-vector distribution hyperparameters which makes it more appropriate when prior knowledge about the data distribution is available. In practice, it is possible to use the Log-likelihood measure for simplicity reasons. For a given noisy test i-vector Y_0, and a noisy i-vector distribution k : $d_k \sim \mathcal{N}(\mu_{Y_k}, \Sigma_{Y_k})$, the distance could be written as:

$$LLK(Y_0, d_k) = -\frac{1}{2} \ln(|\Sigma_{Y_k}|) - \frac{1}{2}((Y_0 - \mu_{Y_k})^{\mathrm{T}} \Sigma_{Y_k}^{-1}((Y_0 - \mu_{Y_k}) - \frac{p}{2} \ln(2\pi) \tag{22}$$

  where p is the dimension of the i-vector space.

10. Recognition performance using the noisy i-vector database

In this section, we first present the system's performance using two different measures (Euclidean distance and LLK) as selection criterion. Then, we inves-

tigate the validity of the method proposed in both clean and noisy enrollment conditions.

In Table 4, we compare the performance given by three systems:

- **i-MAP**: System using I-MAP compensation based on the algorithm described in section 6.
- **Database + i-MAP + LLK measure**: System using the noise distribution database and the Log-likelihood measure as selection criterion.
- **Database + i-MAP + Euclidean distance**: System using the noise distribution database and the Euclidean distance as selection criterion.

The results in Table 4 are given using clean enrollment data and noisy test affected by different noises chosen randomly from {air-cooling, car-driving and crowd-noise} at 4 SNR levels (0dB, 5dB, 10dB and 15dB). Only one noise/SNR level is used for each session. The signal-to-noise ratio threshold used in these experiments is : $SNR_{threshold} = 25dB$.

Table 4: Recognition performance in different matched and mismatched noise and SNR conditions using noisy enrollment data affected by different noises chosen randomly from {air-cooling, car-driving and crowd-noise} at 4 SNR levels (0dB, 5dB, 10dB and 15dB). Only one noise/SNR level is used for each session.

		EER(%)			
		Baseline	**i-MAP**	**Database + i-MAP + LLK**	**Database + i-MAP + Eucl. distance**
Clean enrollment + Noisy test	**0dB**	28.24	14.01	14.10	14.55
	5dB	15.94	6.87	6.93	7.09
	10dB	9.77	3.84	4.01	4.05
	15dB	4.31	2.86	2.88	2.92

It can be clearly seen that the use of the LLK measure as selection criterion produces the lowest equal-error rate when the distribution database is used.

But compared to the baseline performance, the use of the Euclidean distance seems to be adequate if faster computation is required.

Now, we present the recognition performance given by the final system (noise distribution database + i-MAP) in two conditions : clean and noisy enrollment data. For each condition, the results are divided into three sets :

- **Same noise:** Where the noisy data (enrollment or test) are affected by the same noise at different SNR levels.
- **Same SNR:** Where the noisy data (enrollment or test) are affected by different noises at the same SNR level.
- **Heterogeneous setup:** Where all noisy data (enrollment or test) are affected by different noises at different SNR levels.

The signal-to-noise ratio threshold used in the next subsections is $SNR_{threshold} = 25dB$ and the distribution selection is done using the Euclidean distance with respect to the distributions means.

10.1. Recognition performance using clean enrollment data

First, we present in Table 5 the recognition performance using clean enrollment data and different noisy test setups (same SNR and different noises, different noises with the same SNR, different noises and SNR levels). Test segments are affected by {air-cooling, car-driving and crowd-noise}. The SNR level varies between 0dB and 20dB for the *different noise/different SNR* and *same noise* conditions and the added noise is chosen randomly for both *same SNR* and *different noise/different SNR* conditions. Only one noise/SNR level is used for each session.

These results prove the efficiency of our method in mismatched conditions. Compared to the baseline performance, the EER improvement range after the i-MAP compensation is comprised between 32% and 64% while the "multi-style" scoring does not exceed 28% as relative improvement and may even deteriorate the results in certain conditions (restaurant noise).

Table 5: Recognition performance in different test conditions using clean enrollment data. Test segments are affected by {air-cooling, car-driving and crowd-noise}. The SNR level varies between 0dB and 20dB for the *different noise/different SNR* and *same noise* conditions and the added noise is chosen randomly for both *same SNR* and *different noise/different SNR* conditions. Only one noise/SNR level is used for each session.

		EER(%)		
Test condition		**Baseline**	**Multi-style**	**i-MAP**
Same SNR in test	**0dB**	28.24	25.03	14.55
	5dB	15.94	14.57	7.09
	10dB	9.77	8.47	4.05
	15dB	4.31	4.32	2.92
Different noise & SNR in test		15.94	14.30	7.06
Same noise in test	**Car driving**	11.38	8.19	6.80
	Air cooling	20.09	19.56	11.15
	Shopping mall	12.75	10.47	6.15
	Wind	22.53	19.84	7.97
	Restaurant noise	14.12	14.57	6.37

10.2. Recognition performance using noisy enrollment data

Now, we present, in Table 6, the recognition performance using noisy enrollment and test setups (same SNR and different noises in enrollment and test, different noises with the same SNR in enrollment and test, different noises and SNR levels in enrollment and test). Test segments are affected by {air-cooling, car-driving and crowd-noise} and enrollment segments are are affected by {nature noise, rain and engine noise}. The SNR level varies between 0dB and 20dB for the *different noise/different SNR* and *same noise* conditions and the added noise is chosen randomly for both *same SNR* and *different noise/different SNR* conditions. Only one noise/SNR level is used for each session.

The same range of improvement is also observed in this condition. The EER is decreased from 30% up to 62% using i-MAP while the "multi-style" scoring

Table 6: Recognition performance in different matched and mismatched noise and SNR conditions using noisy enrollment data. Test segments are affected by {air-cooling, car-driving and crowd-noise} and enrollment segments are are affected by {nature noise, rain and engine noise}. The SNR level varies between 0dB and 20dB for the *different noise/different SNR* and *same noise* conditions and the added noise is chosen randomly for both *same SNR* and *different noise/different SNR* conditions. Only one noise/SNR level is used for each session.

		EER(%)		
Enrollment and test condition		**Baseline**	**Multi-style**	**i-MAP**
Same SNR in enrollment and test	**0dB**	39.43	35.40	27.75
	5dB	30.54	28.72	12.73
	10dB	17.98	15.26	7.31
	15dB	10.01	7.96	4.09
Different noise & SNR in enrollment and test		30.07	25.28	13.89
Same noise in enrollment and test	**Car driving**	37.12	35.70	21.40
	Air cooling	19.36	16.85	8.88
	Shopping mall	32.08	30.97	13.21
	Wind	29.38	25.87	10.95
	Restaurant noise	38.87	34.87	15.94

does not improve the results by more than 20%.

11. Conclusion

In this work, we introduced an i-vector cleaning technique working only inside the i-vector domain. Our approach assumes that both the clean and noisy i-vector distributions are normally distributed. It allows to estimate the clean i-vector from the noisy one based on both clean i-vectors and noise distributions in the i-vectors space.

Significant improvement was observed using our approach in mismatched conditions compared to a baseline system, a "multi-style" backend system and

a noisy PLDA backend (from 48% up to 64% of relative improvement when used with clean enrollment data). We further showed that i-MAP still reaches high gains in heterogeneous setups (16.27% of EER to be compared with 29.65% of EER for the baseline system) which demonstrates clearly the potential of our approach.

One of the most important steps in the i-MAP compensation scheme is the estimation of the test noise distribution in the i-vector space. Since this process is computationally expensive, we proposed a method to make it much faster based on a noise distribution database pre-computed off-line. This way, we proposed a new algorithm that tries to approximate the test noise distribution by one of the noise distributions present in the database. We showed that while speeding-up the noise distribution estimation process, we still achieved nearly similar gains.

Many extensions to this work could be explored. Further improvement could be brought by using a more complex noise distribution in the i-vector space (such as a mixture of Gaussians) instead of a simple Gaussian. Also, instead of using only a MAP estimation of the clean i-vector given the noisy one, it could be more accurate to integrate the posterior distribution of clean i-vectors in the final scoring phase.

References

[1] A. El-Solh, A. Cuhadar, R. Goubran, Evaluation of speech enhancement techniques for speaker identification in noisy environments, in: Ninth IEEE International Symposium on Multimedia Workshops, ISMW'07., 2007, pp. 235–239.

[2] S. O. Sadjadi, J. H. Hansen, Assessment of single-channel speech enhancement techniques for speaker identification under mismatched conditions., in: Interspeech, 2010, pp. 2138–2141.

[3] C. Hanilçi, T. Kinnunen, R. Saeidi, J. Pohjalainen, P. Alku, F. Ertas, J. Sandberg, M. Hansson-Sandsten, Comparing spectrum estimators in

speaker verification under additive noise degradation., in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 4769–4772.

[4] Y. Lei, L. Burget, N. Scheffer, A noise robust i-vector extractor using vector taylor series for speaker recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 6788–6791.

[5] Y. Lei, M. McLaren, L. Ferrer, N. Scheffer, Simplified vts-based i-vector extraction in noise-robust speaker recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 4037–4041.

[6] D. Martinez, L. Bürget, T. Stafylakis, Y. Lei, P. Kenny, E. Lleida, Unscented transform for ivector-based noisy speaker recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 4042–4046.

[7] P. J. Moreno, B. Raj, E. Gouvea, R. M. Stern, Multivariate-gaussian-based cepstral normalization for robust speech recognition, in: International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 1, 1995, pp. 137–140.

[8] L. Deng, A. Acero, M. Plumpe, X. Huang, Large-vocabulary speech recognition under adverse acoustic environments., in: Interspeech, 2000, pp. 806–809.

[9] M. Afify, X. Cui, Y. Gao, Stereo-based stochastic mapping for robust speech recognition, IEEE Transactions on Audio, Speech, and Language Processing 17 (7) (2009) 1325–1334.

[10] H. Zen, Y. Nankaku, K. Tokuda, Stereo-based stochastic noise compensation based on trajectory gmms, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2009, pp. 4577–4580.

[11] S. Sarkar, K. Sreenivasa Rao, Stochastic feature compensation methods for speaker verification in noisy environments, Applied Soft Computing 19 (2014) 198–214.

[12] C.-H. Lee, On stochastic feature and model compensation approaches to robust speech recognition, Speech Communication 25 (1) (1998) 29–47.

[13] M. Gales, S. J. Young, Hmm recognition in noise using parallel model combination., in: Eurospeech, Vol. 93, 1993, pp. 837–840.

[14] O. Bellot, D. Matrouf, T. Merlin, J.-F. Bonastre, Additive and convolutional noises compensation for speaker recognition., in: Interspeech, 2000, pp. 799–802.

[15] Y. Lei, L. Burget, L. Ferrer, M. Graciarena, N. Scheffer, Towards noise-robust speaker recognition using probabilistic linear discriminant analysis, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 4253–4256.

[16] C. Weng, D. Yu, S. Watanabe, B.-H. F. Juang, Recurrent deep neural networks for robust speech recognition, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 5532–5536.

[17] J. T. Geiger, Z. Zhang, F. Weninger, B. Schuller, G. Rigoll, Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling., in: INTERSPEECH, 2014, pp. 631–635.

[18] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, A. Y. Ng, Recurrent neural networks for noise reduction in robust asr., in: Interspeech, 2012.

[19] X. Feng, Y. Zhang, J. Glass, Speech feature denoising and dereverberation via deep autoencoders for noisy reverberant speech recognition, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 1759–1763.

[20] E. Variani, X. Lei, E. McDermott, I. L. Moreno, J. Gonzalez-Dominguez, Deep neural networks for small footprint text-dependent speaker verification, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 4052–4056.

[21] M. McLaren, Y. Lei, N. Scheffer, L. Ferrer, Application of convolutional neural networks to speaker recognition in noisy conditions., in: Interspeech, 2014, pp. 686–690.

[22] W. Ben Kheder, D. Matrouf, P.-M. Bousquet, J.-F. Bonastre, M. Ajili, Robust speaker recognition using map estimation of additive noise in i-vectors space, in: Statistical Language and Speech Processing, Springer, 2014, pp. 97–107.

[23] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification, IEEE Transactions on Audio, Speech, and Language Processing 19 (4) (2011) 788–798.

[24] P. Kenny, Joint factor analysis of speaker and session variability: Theory and algorithms, CRIM, Montreal,(Report) CRIM-06/08-13.

[25] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, P. Dumouchel, A study of interspeaker variability in speaker verification, IEEE Transactions on Audio, Speech, and Language Processing 16 (5) (2008) 980–988.

[26] D. Matrouf, N. Scheffer, B. G. Fauve, J.-F. Bonastre, A straightforward and efficient implementation of the factor analysis model for speaker verification., in: Interspeech, 2007, pp. 1242–1245.

[27] S. J. Prince, J. H. Elder, Probabilistic linear discriminant analysis for inferences about identity, in: IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007., 2007, pp. 1–8.

[28] N. Brümmer, E. De Villiers, The speaker partitioning problem., in: Odyssey, 2010, p. 34.

[29] P. Kenny, Bayesian speaker verification with heavy-tailed priors., in: Odyssey, 2010, p. 14.

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

[31] The NIST year 2008 speaker recognition evaluation plan, http://www.itl.nist.gov/iad/mig//tests/sre/2008/, [Online; accessed 15-May-2014] (2008).

[32] P.-M. Bousquet, A. Larcher, D. Matrouf, J.-F. Bonastre, O. Plchot, Variance-spectra based normalization for i-vector standard and probabilistic linear discriminant analysis., in: Odyssey, 2012, pp. 157–164.

[33] Freesound.org, http://www.freesound.org.

[34] H. G. Hirsch, FaNT - Filtering and Noise Adding Tool, http://dnt.kr.hsnr.de/download.html, [Online; accessed 15-May-2014].

[35] M.-W. Mak, H.-B. Yu, A study of voice activity detection techniques for nist speaker recognition evaluations, Computer Speech & Language 28 (1) (2014) 295–313.

[36] M. Sahidullah, G. Saha, Comparison of speech activity detection techniques for speaker recognition, arXiv preprint arXiv:1210.0297.

[37] A. Larcher, J.-F. Bonastre, B. G. Fauve, K.-A. Lee, C. Lévy, H. Li, J. S. Mason, J.-Y. Parfait, Alize 3.0-open source toolkit for state-of-the-art speaker recognition., in: Interspeech, 2013, pp. 2768–2772.