The CRIM System for the 2010 NIST Speaker Recognition Evaluation

Patrick Kenny, Pierre Ouellet and Mohammed Senoussaoui

Centre de recherche informatique de Montréal (CRIM)

Patrick.Kenny@crim.ca, Pierre.Ouellet@crim.ca, Mohammed.Senoussaoui@crim.ca

I. Introduction

We attempted all conditions in the evaluation using two architectures both based on the i-vector feature representation of speech segments [1], with 400 dimensional i-vectors for all telephone speech trials (summed as well as 4 wire) and 600 dimensional i-vectors for the microphone trials (microphone-telephone as well as microphone-microphone).

The two architectures are variants of Prince and Elder's Probabilistic Linear Discriminant Analysis [2] which can be viewed as a simplified version of Joint Factor Analysis with a single Gaussian in place of a Gaussian mixture UBM. Our principal innovation consists in using heavy-tailed rather than Gaussian distributions to model speaker and channel effects.

Our modeling algorithms are described in detail in [3]. This paper will be shortly be available at www.crim.ca/perso/patrick.kenny; other well known components of our system such as UBM training, extraction of Baum-Welch statistics and i-vectors are described in other papers on that page so we will only mention these topics in passing in the present document. We will concentrate instead on material that has not otherwise been published, namely our VAD for microphone speech and the development results we obtained in preparing for the evaluation.

We intend to submit results for the extended trial lists when they become available.

II. Signal Processing

A. Features

We use 60-dimensional Gaussianized MFCCs with deltas and delta-deltas, with the Gaussianization performed on all 60 dimensions after adding the deltas and delta-deltas, unlike what we did in our SRE 2008 system.[1] The MFCC were first produced with the HTK tool `HCopy`, and used a window of 25

[1]Thanks to Najim Dehak for persuading us to make this modification.

ms with a 10 ms shift. Energy was normalized, but cepstral mean subtraction was not used (although these two configuration variables should have no effect due to subsequent short-term Gaussianization). Short-term Gaussianization was performed as in [4] on a 301-frame window.

The signal processing for speaker recognition takes less than 2% real-time on a typical CPU (Intel Xeon 2.4 GHz, 8 MB cache).

B. VAD for microphone speech

1) High-level procedure: By VAD, we mean "Voice Activity Detection". We use the term "VAD" instead of "SAD" (Speech Activity Detection), since in our task, speaker recognition, some non-speech voice events such as laughing may be useful for discriminating between speakers.

At the high level, we apply our VAD (described below) to produce a segmentation for each microphone SPHERE file (speaker of interest = channel A only). In the case of interview data, we then subtract from that segmentation the segmentation of channel B (containing the interviewer speech) as provided by NIST in the form of an ASR transcript. The resulting segmentation is then used to cut the desired frames to produce the feature files used for speaker recognition. Note that we do not perform VAD on the interviewer (B) channel: in the SRE 2010 development data, the white noise added to mask the residual interviewee speech is sometimes loud enough to also mask the interviewer speech, thus causing our VAD to detect most of the mask noise as speech, which would lead to the removal of almost all the frames from the audio file.

C. Algorithm

The algorithm is adapted from [5]

a) Initialization feature: We first pass the audio through a Wiener filter, which in our case amounts to a call to the Qualcomm-ICSI-OGI front-end tool `nr` with default parameters [6]. On the resulting filtered audio, a simple feature which we use to initialize our segmentation algorithm is then computed every 10 milliseconds over a window of 25 milliseconds. Given the 8000 Hz sample rate, this amounts to a 80 sample offset and a window of 200 samples. Each window is first normalized to have a zero mean (which corresponds to the ZMEANSOURCE configuration option in HTK). The initialization feature at frame t, $F(t)$, is defined as

$$F(t) = \ln \frac{\delta + E(t)}{\delta + Z(t)},$$

where $E(t)$ is the raw energy over the window at frame t, calulated in the standard way as the sum of squares of samples, and $Z(t)$, the number of zero-crossings, is defined as the number of non-zero samples

such that the preceding sample has the opposite sign. We set $\delta = 10^{-6}$ in order to prevent numerical problems such as division by zero or taking the logarithm of 0.

Note that in [5], the initialization feature is simply mentioned as "energy and relative zero-crossings" mention is not made of how these two are combined. The formula we used aims at favoring lower average frequencies.

b) General VAD procedure: Using internal CRIM tools, we train two GMMs: one for noise and one for speech. The audio is first converted to a sequence of T 10 ms feature vectors of dimension 60, the same as those described in Section II-A, except that Gaussianization is not performed and energy is not normalized. (Note that the initialization feature is calculated with the same frame advance.)

From the 60 dimensional feature vectors, we select H frames for which the initialization feature has the highest values, and train an initial speech GMM from these frames. Similarly, we select L frames for which the initialization feature has the lowest values to train an initial noise GMM. In both cases, the EM algorithm is used (as opposed to MAP in [5]), with three iterations. The first iteration re-estimates the mean vectors only and the second and third iteration re-estimate the mean vectors, diagonal covariance matrix, and component weights.

An initial Viterbi alignment of the whole sequence of T feature vectors is then produced using the initial GMMs as models. We then train a new noise and a new speech model from the resulting segments using all of the frames (unlike the subsets of size H and L used in the initialization step). We repeat this train/align procedure until the noise/speech ratio of the output from two consecutive iterations varies by less than 1% (as per [5]) or until 20 iterations have been performed, whichever comes first.

c) The VAD procedure for interview data: In this case, we use the same parameters as in [5]: we define H as $0.1T$ and L as $0.2T$. The noise GMM is trained to have 4 Gaussians, and the speech GMM 16 Gaussians.

d) The VAD procedure for microphone-recorded phone conversations: In this case (phonecall/mic), we use $L = 0.9T$, and 16 Gaussians for the noise model, and the remaining configuration is the same as for the interview data. Empirically, we determined this to give better results than the interview configuration, which tends to assign too many non-speech frames as speech. Our motivation was to bias the configuration so more reasonable segmentations would be produced. The percentage of speech from the speaker of interest is generally lower than in the phonecall/mic case than in the interview case.

e) Advantages of our VAD procedure: The CRIM-VAD procedure is self-normalizing, as is the one it is derived from [5]. Where simpler VAD software may require the setting of energy thresholds, our procedure works even for audio files where the volume is quite low. In those cases, of course, the

dynamic range is reduced, but our method copes well with that.

There are cases, such as empty audio files, or audio files full of a loud background noise, which cause the procedure to fail because no data is found either for the noise model or the for speech model, that is, all the frames are assigned to one or the other. This oversight in the design turned out to be useful in practice for detecting such suspect cases.

f) Disadvantages of our VAD procedure: The procedure is compuatationally costly: about 10% real-time on average. Most of the time is spent in the align/train loop. Also, the procedure can not be applied on-the-fly: the audio for a complete utterance must be available beforehand.

III. Extraction of Baum-Welch statistics and i-vectors

Baum-Welch statistics and the 400 dimensional i-vectors that we used for telephone trials are extractd as in [1]. For microphone trials (mic-tel as well as mic-mic) we augmented the i-vector dimension to 600 as described in [7]. Extracting Baum-Welch statistics takes 4% real-time. In the 600 dimensional case i-vectors are generated at the rate of 2,500 per hour using 3.3 Gb of RAM. (This is the only memory intensive computation in our system.)

IV. Enrollment and recognition

The algorithms are described in [3]. There is no enrollment procedure as this term is usually understood but comparing an enrollment segment with a test segment is computationally expensive as it involves iterating a Variational Bayes algorithm to convergence in order to calculate the likelihood ratio needed to make verification decisions. (A disadvantage of heavy-tailed modeling compared to Gaussian modeling is that in the latter case, the Variational Bayes algorithm converges in a single iteration.)

Except for microphone trials (that is, mic-tel and mic-mic) and some trials involving 10 second test segments, we did not use score normalization. In the other cases we used s-norm [3]. For each gender, we used two s-norm cohorts of size 200–250, one for telephone imposters and the other for microphone imposters.

In the tel-tel case (without score normalization) we can process trials at the rate of about 2.5 million per day; in the mic-mic and mic-tel cases we can only process trials at the rate of 0.5 million per day.

The scores that we have submitted are not intended to be interpreted as calibrated log likelihood ratios, except in the case of the core condition.[2]

[2]Thanks to Niko Brummer who did the calibration for us.

V. Results obtained during development

We tested our system using the 2008 SRE trial lists in situations where the 2010 DCF is the same as for 2008; in the other situations we tested using extended trial lists consisting of several million trials in all.

We used English language female trials for development as our experience has almost invariably been that female trials are harder than males. We found that for the 2008 SRE trial lists, the results for male trials are uniformly better in all of the 4 wire telephone conditions and in the core mic-tel and mic-mic trials. Trials involving summed data were the only clear exception to this rule. We have evidence that this may be due to gender assignment errors in the the trial lists.[3]

A. Results obtained with the 2010 DCF

Table I summarizes results that we obtained with extended trial lists intended to similuate the 2010 core-core and 8conv-core conditions which will be evaluated using the new detection cost function. NDCF indicates the the normalized detection cost function described in the evaluation plan. Concerning the NDCF value of 0.445 for the tel-tel trials, we note that this figure reduces to 0.31 after removing some trials which are non-targets according to the 2008 key but which actually appear to be target trials.

TABLE I

EER, 2008 DCF and 2010 NDCF extended trial lists, female targets only

	targets	non-targets	EER	2008 DCF	2010 NDCF
core tel-tel	798	1,940,902	2.0%	0.007	0.445
core mic-tel	31,204	1,149,624	3.3%	0.015	0.563
core mic-mic	14,770	720053	2.6%	0.014	0.543
8conv-core	460	361,681	0.65%	0.002	0.093

The tel-tel trial list was provided by Agnitio. We gratefully acknowldge the effort they expended to track down erroneous gender assignments and false non-target trials (target speakers with multiple PINs). The mic-tel and mic-mic trial lists were kindly provided by BUT and MIT. We built our own list for the 8conv-core condition.

[3]Thanks to Fabio Castaldo for tracking these down.

B. Results obtained on 4 wire telephone trials with the 2008 DCF

We used the NIST 2008 trial lists for these tests. The results are summarized in Table II. Results for short2-short3 and 8conv-short3 are included for completeness. We used s-norm score normalization for the short2-10sec and 10sec-10sec conditions but not for any of the other conditions.

TABLE II

EER and 2008 DCF on 4 wire telephone conditions, 2008 trial lists, female targets only

	EER	2008 DCF
8conv-10sec	3.2%	0.012
8conv-short3	1.1%	0.004
short2-short3	2.1%	0.008
10sec-10sec	9.5%	0.045
short2-10sec	4.2%	0.021

C. Results obtained on trials involving summed telephone data with the 2008 DCF

We used the NIST 2008 trial lists for these tests and we did not use score normalization. The results are summarized in Table III

The summed data was diarized using the algorithm in [8].

For the the 3summed train conditions, we performed speaker clustering on the the 3summed data using an exhaustive search to optimize the likelihood criterion in [3]. We used the same stragtegy for the 8summed train conditions in 2010.

Cross-gender trials are a potential source of difficulty in the summed tests but we did not attempt to address this issue. In situations where the the NIST .ndx file indicated that a given target speaker was male, we used our male heavy-tailed model to calculate likelihood ratios without regard to the gender of the test-segment speaker and similarly for females.

REFERENCES

[1] N. Dehak, R. Dehak, P. Kenny, N. Brummer, P. Ouellet, and P. Dumouchel, “Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification,” in *Proc. Interspeech*, Brighton, UK, Sept. 2009.

[2] S. J. D. Prince and J. H. Elder, “Probabilistic linear discriminant analysis for inferences about identity,” in *Proc. 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[3] P. Kenny, “Bayesian speaker verification with heavy tailed priors,” in *Proc. Odyssey 2010: The speaker and Language Recognition Workshop*, Brno, Czech Rebublic, June 2010.

TABLE III

EER and 2008 DCF on summed telephone conditions, 2008 trial lists, female targets only

	EER	2008 DCF
short2-summed	2.7%	0.015
3summed-summed	2.5%	0.010
8conv-summed	1.5%	0.007
3summd-short3	2.2%	0.007

[4] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. Speaker Odyssey*, Crete, Greece, June 2001, pp. 213–218.

[5] H. Sun *et al.*, "Speaker diarization for meeting room audio," in *Proc. Interspeech 2009*, Brighton, UK, Sept. 2009.

[6] A. Adami *et al.*, "Qualcomm-ICSI-OGI Features for ASR," in *Proc. ICSLP 2002*, Denver, CO, Mar. 2002.

[7] M. Senoussaoui, P. Kenny, N. Dehak, and P. Dumouchel, "An i-vector extractor suitable for speaker recognition with both microphone and telephone speech," in *Proc. Odyssey Speaker and Language Recogntion Workshop*, Brno, Czech Republic, June 2010.

[8] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE J. of Selected Topics in Signal Processing*, Dec. 2010. [Online]. Available: http://www.crim.ca/perso/patrick.kenny