1. INTRODUCTION

We submit three different systems to participate in the fixed training condition. We designate System 3 as the primary system.

The distinctive features of our systems are the novel use of Joint Bayesian scoring and ladder network based feature learning. Due to limited time, we use a simple front-end - the DNN used for i-vector extractor is trained using only clean data. Multi-condition training should greatly improve the front-end.

2. SUBMITTED SYSTEMS

Our speaker recognition system consists of three main modules, which are i-vector extractor based on Deep Neural Networks (DNNs), i-vector post-processing (length normalization) and discriminant analysis. We submit three different systems which are described briefly as follows. The system flowchart is shown in Figure 1, and the training data statistics in system building are summarized in Table 1. The performance of the three systems on the SRE 2016 “dev” set (i.e. LDC2016E46-SRE16-CallMyNet-Training-Data labeled part) is reported in Table 3. It can be seen that the combined System 3 gives the best performance on this "dev" set.

2.1. System 1

2.1.1. i-vector extractor and post-processing

System 1 uses DNN-based i-vector extractor. The acoustic features used are 40-dimensional Mel-Frequency Cepstral Coefficients (MFCCs), including 20-dimensional static features and first-order derivatives. A DNN is trained on Fisher data with 5 hidden layers and 5335 senones. The input of the DNN is the MFCCs extracted using 21 frames (11 frames before and 9 frames after). The i-vector dimension is 600 and the i-vector extractor is trained by the EM algorithm. The extracted i-vectors are post-processed by length normalization to \( \sqrt{600} \). The above steps are mostly conducted using Kaldi toolkit [1].

2.1.2. Discriminant analysis: Joint Bayesian

Joint Bayesian (JB) is used in our system for discriminant analysis, which is originally proposed in [2] for face verification and further developed in [3] for speaker verification. In JB, i-vectors are modeled as two independent Gaussian-s which represent speaker identity and intersession residuals respectively. The \( j \)-th i-vector of speaker \( i \), denoted by \( x_{ij} \in \mathbb{R}^d \), is decomposed as:

\[
 x_{ij} = \mu_i + \varepsilon_{ij} \tag{1}
\]

where \( \mu_i \sim \mathcal{N}(0, S_{\mu}) \) is the speaker identity variable, \( \varepsilon_{ij} \sim \mathcal{N}(0, S_{\varepsilon}) \) models the within-speaker variability. The model parameters are \( \Theta = \{S_{\mu}, S_{\varepsilon}\} \) which is estimated by the EM algorithm through iteratively optimizing the expected complete log-likelihood function.

In speaker verification testing, we calculate the log-likelihood ratio (LLR) as the score to determine whether one set of enroll i-vectors \( x_1 \) (including one or three i-vectors) and the test i-vector \( x_2 \) are from the same speaker

\[
 LLR(x_1, x_2) = \log p(x_1, x_2) - \log p(x_1) - \log p(x_2) \tag{2}
\]

2.2. System 2

System 2 inherits the i-vector extractor module from System 1. In System 2, we propose to use ladder networks [4, 5] with center loss [6] to non-linearly extract 500-dim features from the 600-dim i-vectors generated in System 1. Then Fisher L-DA is applied, which makes the features more like Gaussians. Finally, JB-based scoring is used to generate the verification score as in Equ. (2).

The ladder networks [5] are initially designed for unsupervised feature learning and are extended in [4] for semi-supervised feature learning. For our System 2, we train a ladder network using both speaker labeled i-vectors (Switchboard-2 and SRE which basically are out-of-domain data for evaluation) and unlabeled i-vectors (a subset from LDC2016E46-SRE16-CallMyNet-Training-Data unlabeled part which are more matched to the evaluation data). The motivation is to leverage unlabeled in-domain data.
In addition to using the re-construction errors, the original ladder networks only consider the categorical errors, which are suitable for closed-set identification. For verification (open-set inherently), the features need to be not only separable but also discriminative, so that both compact intra-class variations and separable inter-class differences can be achieved. The categorical loss objective only encourages the separability of features that are not sufficient for verification. Therefore we further add center loss [6] to reduce intra-class variations to learn more discriminative features.

### 2.3. System 3: System combination

The scores of System 3 are the combination of LLRs from System 1 and System 2. The final scores are calculated as

$$ Score = \alpha \cdot LLR_{jb} + (1 - \alpha) \cdot LLR_{ladder} \quad (3) $$

where $\alpha$ is the interpolation coefficient which is tuned using LDC2016E46-SRE16-CallMyNet-Training-Data labeled part.

### 3. TIMING REPORT

In Table 2, we show the running time statistics for main modules in our systems. We differentiate the use of CPUs or GPUs.

In Table 2, processing trials from SAD to i-vector extraction consists of performing SAD, extracting features, extracting statistics and extracting i-vectors. According to Table 2, the single-threaded CPU time and memory used to process an audio file (whether an enrollment file or a test file) are about 6 hours * 10 / 10496 = 20 seconds and 7 G respectively.

Processing trials after i-vector extraction consists of ladder-network based feature transformation (System 2) and JB scoring. According to Table 2, the single-threaded CPU time and memory used to process a trial after obtaining i-vectors for the enrollment file and the test file, are about 10 minutes / 1986728 = 0.3 ms and 2 G respectively.

### Table 3. The performance of the three systems on SRE 2016 "dev" and "eval" set. The only difference from our submission result is that here we applied TNorm to system scores, which significantly improve the performance.
4. REFERENCES


