INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Robust online i-vectors for unsupervised adaptation of DNN acoustic models: A study in the context of digital voice assistants

Harish Arsikere, Sri Garimella

Amazon.com, Bangalore, India
`arsikere@amazon.com, srigar@amazon.com`

Abstract

Supplementing log filter-bank energies with i-vectors is a popular method for adaptive training of deep neural network acoustic models. While *offline* i-vectors (the target utterance or other relevant adaptation material is available for i-vector extraction prior to decoding) have been well studied, there is little analysis of *online* i-vectors and their robustness in multi-user scenarios where speaker changes can be frequent and unpredictable. The authors of [1] showed that online adaptation could be achieved through segmental i-vectors computed using the hidden Markov model (HMM) state alignments of utterances decoded in the recent past. While this approach works well in general, it could be rendered ineffective by speaker changes. In this paper, we study robust extensions of the ideas proposed in [1] by: (a) updating i-vectors on a per-frame basis based on the incoming target utterance, and (b) using lattice posteriors instead of one-best HMM state alignments. Experiments with different i-vector implementations show that: (a) when speaker changes occur, lattice-based frame-level i-vectors provide up to 6% word error rate reduction relative to the baseline [1], and (b) online i-vectors are more effective, in general, when the microphone characteristics of test utterances are not seen in training.

Index Terms: i-vectors, online, frame-level, lattice posteriors, DNN adaptation

1. Introduction

A typical deep neural network (DNN) acoustic model (AM) is trained to output posterior probabilities of senones (tied context-dependent HMM states) using log filter-bank energies (LFBEs) as input [2]. To alleviate the effects of speaker and channel variability, most present-day automatic speech recognition (ASR) systems use some form of DNN AM adaptation. Broadly speaking, adaptation can be implemented either through feature- or model-space approaches [3–7], or by simply supplementing LFBEs with speaker- and/or channel-related information [1,8–12]. The latter approach, dominated by the use of i-vectors, is widely adopted owing to its simplicity and state-of-the-art performance in unsupervised settings.

Conceptually, an i-vector simply specifies how the mean supervector of a 'generic' Gaussian mixture model (GMM) must be transformed in a low-dimensional subspace in order to best represent an observed speech segment. In the context of digital voice assistants, a speech segment, or utterance, can be defined as the signal recorded after the device starts listening to the user and before the end-pointer declares end of speech. The i-vector estimation procedure for a speech segment depends on several practical considerations:

- Access to target utterance: Offline i-vectors are computed either by using the entire target utterance, i.e., the utterance that needs to be decoded, or by using adaptation material that is relevant to the target speaker or environment (which is known beforehand). Online i-vectors, on the other hand, are purely causal in nature and do not make use of any prior information. Clearly, online i-vectors are better suited for real-time decoding.
- Update frequency: One can estimate i-vectors on a segmental or per-frame basis depending on how frequently the sufficient statistics are updated. By definition, segmental i-vectors are constant across all the frames of a given segment.
- Sufficient statistics accumulation: Sufficient statistics can be accumulated using the Gaussians of a universal background model (UBM) or the state-tied GMM-HMM of an ASR system. The association between feature vectors and Gaussians can be obtained using: (a) UBM posteriors, (b) DNN AM posteriors or (c) HMM state alignments.

Most studies, with the exception of [1,12], report adaptation results using segmental offline i-vectors extracted from UBM-GMM or DNN AM posteriors [9–11]. Although tools like Kaldi [13] support frame-level online i-vector extraction, a thorough comparison of segmental and frame-level online i-vectors, particularly in multi-user scenarios where speaker changes can be frequent and unpredictable, is missing. Therefore, in this paper, we compare different online i-vector implementations and analyze their robustness to speaker switches—which can happen often when two or more people of a household interact with the same device (e.g., the Amazon Echo or Google Home) on a day-to-day basis. We also analyze the behavior of online i-vectors in situations where the microphone characteristics of the test data are not seen in training.

The authors of [1] showed that online adaptation could be achieved via segmental i-vectors extracted from the HMM state alignments of previous utterances; they used exponential decay to bias the sufficient statistics to the most recent utterance history. It was also shown that for accumulating i-vector statistics, HMM state alignments obtained from an ASR system are better than UBM or DNN posterior scores. While the implementation of [1] works well in general, it could be rendered ineffective by speaker changes. Let us consider an example:

History*: Interaction between a female speaker and the device*
Utterance1*: Speech from a male speaker*
Response from the device
Utterance2*: Speech from the same male speaker*

If a segmental i-vector is computed from the *History* (as in [1]), it would be suboptimal for decoding *Utterance1* because of the female-to-male transition. If *Utterance1* is short, the *History* for *Utterance2* will not change much thereby affecting its decoding to some extent. Also, the effect of the above speaker transition could be more pronounced if the HMM state alignments of the *History* are unreliable, e.g., due to background noise.

In this paper, we study two robust extensions to the implementation of [1]: computing frame-level i-vector updates based on the incoming (partial) target utterance, and using lattice pos-

http://dx.doi.org/10.21437/Interspeech.2017-1342

teriors instead of HMM state alignments. The former extension provides better adaptation to the target speaker or environment, while the latter extension tries to mitigate the effect of incorrect alignments. Since online frame-level updates cannot make use of alignments or lattices for the target utterance, DNN AM posteriors are used to accumulate statistics on the incoming frames; the best posterior is a substitute for alignments, while the top K posteriors are a substitute for lattices.

We first present a Bayesian view of i-vectors by explicitly specifying the latent variables and simplifying assumptions that lead to the standard i-vector estimate of [14]. This is one of our key contributions here given that seminal studies [14, 15] have approached i-vector theory from a joint factor-analysis perspective. Also, our Bayesian view offers an easy framework for formalizing segmental and frame-level online i-vectors.

2. A Bayesian view of i-vectors

Let $\theta = \{w_i, \mu_i, \mathbf{\Sigma}_i\}_{i=1}^{M}$ be the set of known parameters of a GMM with M components, where w_i, μ_i and $\mathbf{\Sigma}_i$ denote the weight, mean and covariance, respectively, of the i^{th} Gaussian component; θ can correspond to the Gaussians of a UBM [14] or a state-tied GMM-HMM system. If μ is the GMM mean supervector formed by concatenating the component means, and $\mathbf{T}$ = $\left[\mathbf{T}_1; \mathbf{T}_2; \ldots; \mathbf{T}_M\right]$ (";" indicates vertical stacking) is a matrix whose column space forms a subspace of the GMM supervector space, then $\mu + \mathbf{Tq}$ forms an affine subspace in which each point corresponds to a unique 'adapted' GMM depending on the value of $\mathbf{q}$. Let $\{\theta, \mathbf{T}\}$ be denoted by Θ.

If a set of feature vectors $\mathbf{X}(n)$ = $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ is generated by the above entities, the following latent random variables are associated with it: $\mathbf{q}$, which specifies a point in the affine subspace $\mu + \mathbf{Tq}$, and $\mathbf{Z}(n)$ = $[\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n]$, where $\mathbf{z}_t$ specifies the Gaussian component (of the GMM specified by $\mathbf{q}$) responsible for generating $\mathbf{x}_t$. The latent variable $\mathbf{q}$ is assumed to have a standard normal prior $\mathcal{N}(.; \mathbf{0}, \mathbf{I})$, and $\mathbf{z}_t$ $(1 \leq t \leq n)$ is modeled as a multinomial random variable with one trial and M categories, with probability of success for the i^{th} category being w_i; $\mathbf{z}_t$ can be written as a boolean vector $\left[z_{t1}; z_{t2}; \ldots; z_{tM}\right]$ where all entries are zero except for the one that corresponds to the Gaussian component responsible for $\mathbf{x}_t$.

During inference using this Bayesian framework (note that $\mathbf{T}$ is assumed to be known and its estimation is not discussed here), one needs to determine the joint posterior density of the latent variables $\{\mathbf{Z}(n), \mathbf{q}\}$, given the observed features $\mathbf{X}(n)$. While this computation is mathematically possible, it is not feasible in practice owing to the lack of efficient methods. An approximation that simplifies this inference leads to the standard i-vector estimate of [14].

The simplifying approximation is that $\mathbf{z}_t$ $(1 \leq t \leq n)$ is *treated as an observed variable*. Although not directly observable, a good guess for $\mathbf{z}_t$ could be used as its observed value. For example, the guess could be set to the Gaussian component that maximizes the value of $f_\Theta(\mathbf{z}_t | \mathbf{x}_t, \mathbf{q} = \mathbf{0})$. This is typically achieved in practice through UBM or DNN AM posterior scores (and more recently though GMM-HMM state alignments). With this approximation, our inference simplifies to estimating the posterior density of $\mathbf{q}$ given $\{\mathbf{X}(n), \mathbf{Z}(n)\}$.

Using the known parameters Θ and the 'observed' variables $\mathbf{Z}(n)$, the log posterior density $\log f_\Theta(\mathbf{q}|\mathbf{X}(n), \mathbf{Z}(n))$ can be shown to be proportional to:

$$\mathbf{q}^T \left[\mathbf{I} + \sum_{i=1}^{M} \gamma_i \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{T}_i\right] \mathbf{q} - 2\mathbf{q}^T \left[\sum_{i=1}^{M} \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{f}_i\right], \quad (1)$$

where γ_i and $\mathbf{f}_i$ are given by Eq. (2):

$$\gamma_i = \sum_{t:z_{ti}=1} 1 \qquad \mathbf{f}_i = \sum_{t:z_{ti}=1} (\mathbf{x}_t - \mu_i). \quad (2)$$

Based on the quantity in Eq. (1), the posterior density of $\mathbf{q}$ given $\{\mathbf{X}(n), \mathbf{Z}(n)\}$ can be identified as $\mathcal{N}(.; \mu_\mathbf{q}, \mathbf{\Sigma}_\mathbf{q})$, where:

$$\mathbf{\Sigma}_\mathbf{q} = [\mathbf{I} + \mathbf{S}_0]^{-1} = \left[\mathbf{I} + \sum_{i=1}^{M} \gamma_i \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{T}_i\right]^{-1}$$

$$\mu_\mathbf{q} = \mathbf{\Sigma}_\mathbf{q} \mathbf{S}_1 = \mathbf{\Sigma}_\mathbf{q} \left[\sum_{i=1}^{M} \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{f}_i\right]. \quad (3)$$

In Eq. (3), $\mathbf{S}_0$ and $\mathbf{S}_1$ denote the zeroth- and first-order 'partial sums,' respectively, and $\mu_\mathbf{q}$ is the estimated i-vector.

3. Online i-vectors for DNN adaptation

The notation here is mostly borrowed from Section 2. Since online i-vectors are causal, we use $\mathbf{X}(0) = [\ldots, \mathbf{x}_{-1}, \mathbf{x}_0]$ to denote frames that are already seen by a particular device (i.e. all previously decoded utterances). $\mathbf{Z}(0)$ can be similarly defined. Additionally, we use $\mathbf{X}(l)$ = $[\mathbf{x}_1, \ldots, \mathbf{x}_l]$ to denote frames from the incoming (partial) target utterance. The interpretation of γ_i, $\mathbf{f}_i$, $\mu_\mathbf{q}$ and $\mathbf{\Sigma}_\mathbf{q}$ would also change slightly as we will see below.

3.1. Segmental online i-vectors

As mentioned in Section 1, a segmental i-vector is held constant for the entire target utterance. In [1], segmental online i-vectors were computed using the sufficient statistics given by Eq. (4). Since $\mathbf{X}(0)$ includes all the previously-seen utterances, an exponential decay factor τ (> 0) was used to emphasize the most recent speech frames. The values of z_{ti} $(t \leq 0)$ were obtained from the 1-best HMM alignments of the decoder.

$$\gamma_i(0) = \sum_{t:z_{ti}=1, t\leq 0} e^{\tau t}$$

$$\mathbf{f}_i(0) = \sum_{t:z_{ti}=1, t\leq 0} e^{\tau t} (\mathbf{x}_t - \mu_i) \quad (4)$$

Since 1-best HMM alignments might be unreliable at times, e.g., due to background noise or long medial pauses, we propose to use senone posteriors from the decoding lattice instead. The sufficient statistics in this case are computed as:

$$\gamma_i(0) = \sum_{t\leq 0} e^{\tau t} P_{lat}(i; t)$$

$$\mathbf{f}_i(0) = \sum_{t\leq 0} e^{\tau t} P_{lat}(i; t) (\mathbf{x}_t - \mu_i), \quad (5)$$

where $P_{lat}(i; t)$ denotes the lattice posterior score for senone i at time t. Once $\gamma_i(0)$ and $\mathbf{f}_i(0)$ are available, the partial sums $\mathbf{S}_0(0)$ and $\mathbf{S}_1(0)$ are computed using Eq. (6):

$$\mathbf{S}_0(0) = \sum_{i=1}^{M} \gamma_i(0) \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{T}_i$$

$$\mathbf{S}_1(0) = \sum_{i=1}^{M} \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{f}_i(0), \quad (6)$$

and the i-vector $\mu_\mathbf{q}(0)$ is computed from Eq. (7):

$$\mu_\mathbf{q}(0) = [\mathbf{I} + \mathbf{S}_0(0)]^{-1} \mathbf{S}_1(0). \quad (7)$$

3.2. Frame-level online i-vectors

To update sufficient statistics using the incoming (partial) target utterance $\mathbf{X}(l)$, one cannot use 1-best HMM alignments or lattice posteriors because they would not be available in real-time decoding scenarios. However, DNN AM posteriors can be used since they are computed on the fly during decoding. Eq. (8) is used to accumulate statistics from $\mathbf{X}(l)$:

$$\gamma_i(l) = \sum_{t=1}^{l} e^{-\tau(l-t)} P_{dnn}(i|\mathbf{x}_t)\, \mathbb{I}_{ti}^{K}$$

$$\mathbf{f}_i(l) = \sum_{t=1}^{l} e^{-\tau(l-t)} P_{dnn}(i|\mathbf{x}_t)\, \mathbb{I}_{ti}^{K}(\mathbf{x}_t - \mu_i), \tag{8}$$

where $P_{dnn}(i|\mathbf{x}_t)$ denotes the DNN posterior score for senone i given the feature vector $\mathbf{x}_t$, and $\mathbb{I}_{ti}^{K}$ is an indicator function (1 or 0) for whether or not $P_{dnn}(i|\mathbf{x}_t)$ is among the top K posterior scores. K can take any value between 1 and the number of senones, but it should be set to a small value (10–20) to achieve computational efficiency in real-time decoding and to avoid accumulating statistics on the less probable senones. (If the DNN is trained well, the top few posteriors are expected to include the 'correct senone.') $\mathbb{I}_{ti}^{K}$ is not used with lattice posteriors (Eq. (5)) because decoding lattices are usually sparse enough for statistics to be accumulated over all the 'active' senones. Using $\gamma_i(l)$ and $\mathbf{f}_i(l)$, the zeroth- and first-order partial sums are computed from Eq. (9):

$$\mathbf{S}_0(l) = \mathbf{S}_0(0)\, e^{-\tau l} + \sum_{i=1}^{M} \gamma_i(l)\, \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{T}_i$$

$$\mathbf{S}_1(l) = \mathbf{S}_1(0)\, e^{-\tau l} + \sum_{i=1}^{M} \mathbf{T}_i^T \mathbf{\Sigma}_i^{-1} \mathbf{f}_i(l), \tag{9}$$

and the i-vector $\mu_{\mathbf{q}}(l)$ is computed from Eq. (10):

$$\mu_{\mathbf{q}}(l) = [\mathbf{I} + \mathbf{S}_0(l)]^{-1}\, \mathbf{S}_1(l). \tag{10}$$

In Eq. (9), $\mathbf{S}_0(0)$ and $\mathbf{S}_1(0)$ are the partial sums accumulated from $\mathbf{X}(0)$ (Eq. (6))—they can be computed from 1-best HMM alignments or lattice posteriors. Note that $\mu_{\mathbf{q}}(l)$ can be updated on a per-frame basis, i.e., whenever a new frame of the target utterance is available. Once the target utterance is decoded, $\mathbf{X}(0)$, $\mathbf{S}_0(0)$ and $\mathbf{S}_1(0)$ are updated to form the 'history' for the next target utterance.

4. System description

4.1. Corpora

In this study, we use data collected from two device types with different microphone characteristics. Let the device types be denoted by *DType1* and *DType2* for convenience. Data collected from both device types consists of utterances from several device units. All recordings are sampled at 16 kHz.

The training and validation sets are of *DType1*. The training set is a standard internal collection that is used for experimental purposes. About 10% of the training data (randomly chosen) is used for DNN pre-training.

Two test sets are used in our experiments: *Test1*, of *DType1*, and *Test2*, of *DType2*. *Test1* and *Test2* contain 102k and 116k utterances, respectively. Since *DType2* is not part of the training set, *Test2* helps us assess the efficacy of online i-vector adaptation for 'unseen' device types. The training, validation and test sets do not have any device-unit or speaker overlap.

4.2. LFBE extraction

LFBE features are extracted every 10 ms using analysis frames of duration 25 ms. For each frame, the power spectrum (squared magnitude of the discrete Fourier transform) is integrated using 64 Mel-warped filters, and the resulting filter-bank outputs are transformed using natural logarithm to obtain LFBEs. LFBEs are normalized by subtracting the time-varying mean estimate obtained through an autoregressive update [16].

4.3. GMM-HMM AM and T-matrix estimation

An HMM AM with single-Gaussian states is trained for extracting i-vectors. Specifically, each context-dependent tied state or senone (which is specified by the acoustic decision tree) is modeled using a single diagonal-covariance Gaussian. To associate a feature vector with one or more senones of this single-Gaussian HMM AM, one can use alignments, lattices or DNN posteriors. Using a susbet of our training data, the Gaussians are estimated from 40 dimensional features which are obtained by stacking 9 LFBE frames (current frame plus a left and right context of 4), and applying a block diagonal discrete cosine transform (DCT), a linear discriminant analysis (LDA) transform and a maximum likelihood linear transform (MLLT) [17]. The LDA and MLLT transforms are obtained from an in-house ASR system.

A $\mathbf{T}$ matrix is trained to estimate 32-dimensional i-vectors (using the same training subset as above). The matrix is initialized randomly and trained for 12 epochs using the expectation-maximization (EM) algorithm ($\mathbf{T}$-matrix estimation details can be found in [15]).

4.4. Online i-vector extraction

For DNN training, utterance-level online i-vectors (Eqs. (4), (6), (7)) are extracted using the forced alignments obtained based on reference transcripts (note that this is the implementation used in [1]). Test online i-vectors are extracted in different ways, although they are all decoded using the same DNN:

$\boldsymbol{\mathcal{A}}|\boldsymbol{\cdot}$: alignment based | utterance (segmental) update [1]
$\boldsymbol{\mathcal{L}}|\boldsymbol{\cdot}$: lattice based | utterance (segmental) update
$\boldsymbol{\mathcal{A}}|\boldsymbol{b}$: alignment based | frame update: best DNN posterior
$\boldsymbol{\mathcal{L}}|\boldsymbol{b}$: lattice based | frame update: best DNN posterior
$\boldsymbol{\mathcal{A}}|\boldsymbol{p}$: alignment based | frame update: top K DNN posteriors
$\boldsymbol{\mathcal{L}}|\boldsymbol{p}$: lattice based | frame update: top K DNN posteriors

While one might argue that it is best to match train and test i-vectors, our current setup is meaningful because the above i-vector implementations are conceptually the same, differing just in statistics accumulation methods and update frequencies. We in fact experimented with DNNs trained on lattice-based frame-level online i-vectors, but the results were not very different in comparison to the present training strategy.

The forced alignments that are required for i-vector extraction during training are obtained using an in-house HMM-DNN ASR model (this model has the same acoustic decision tree and HMM structure as the GMM-HMM described in Section 4.3). For test i-vector extraction, 1-best alignments, lattice posteriors and DNN posteriors are obtained from the HMM-DNN model used for decoding (i.e. the DNN trained for this study). For all i-vector estimates, the exponential decay factor τ is set to 0.002; this corresponds to an effective time window of 500 frames or 5 seconds. Note that i-vector statistics are not accumulated on the silence and non-speech senones. For $\boldsymbol{\mathcal{A}}|\boldsymbol{p}$ and $\boldsymbol{\mathcal{L}}|\boldsymbol{p}$, the value of K is set to 10.

4.5. DNN training

The DNN takes LFBEs and i-vectors as inputs, and outputs posterior probabilities for the senones specified by the acoustic decision tree. LFBEs and i-vectors are mean and variance normalized based on the global statistics of the training data, prior to

DNN training. LFBEs from 17 frames (the current frame plus a left and right context of 8 frames) are stacked to form one set of DNN inputs. The 32-dimensional i-vectors are passed through a non-linear bottleneck layer with 16 sigmoid units; based on the findings in [1], this i-vector specific hidden layer is essential for good adaptation performance. The stacked LFBEs are concatenated with the output of the i-vector specific hidden layer to form the inputs to the first hidden layer of the DNN. The DNN has 8 non-linear hidden layers with 1664 sigmoid units each, and an output softmax layer.

Frame targets for DNN training are obtained by force aligning reference transcripts with our in-house ASR system (mentioned in Section 4.4). The DNN is discriminatively pre-trained (one layer at a time) by minimizing the standard cross-entropy loss function. The full network is then cross-entropy trained using stochastic gradient descent. Beginning with an initial value of 0.008, the learning rate is exponentially reduced by epoch for 14 epochs. The mini-batch size is chosen to be 256. For boosted maximum mutual information (BMMI) sequence training [18], the best cross-entropy trained model is selected using the validation set—this model is used to obtain the required numerator alignments and denominator lattices. The learning rate, acoustic scaling factor and lattice boosting factor are 1e-5, 0.08 and 0.1, respectively. The DNN is trained for 2 epochs using the BMMI loss function.

5. Experimental results

For all experiments, we report relative word error rate (WER) reductions with respect to segmental alignment-based i-vectors ($\boldsymbol{\mathcal{A}}|\cdot$) [1]. Compared to an *LFBE-only* (i.e. without i-vectors) DNN which is trained exactly as described in Section 4.5, the *LFBE+i-vector* DNN, when provided with $\boldsymbol{\mathcal{A}}|\cdot$, provides a relative WER reduction of 3.2 and 7.1% for *Test1* and *Test2*, respectively. This confirms that $\boldsymbol{\mathcal{A}}|\cdot$ is effective for online adaptation, thus making it a valid baseline for our experiments.

Based on the above baseline results, it is important to note that adaptation is more effective for *Test2* than for *Test1* (7.1% vs. 3.2% WER reduction due to adaptation). This indicates that i-vectors can compensate for mismatch in microphone characteristics, in addition to encoding speaker and channel information; this result is one of the key findings of this study.

Tables 1 and 2 show the relative WER reductions (achieved by different i-vector implementations with respect to $\boldsymbol{\mathcal{A}}|\cdot$) for *Test1* and *Test2*, respectively. Since speaker (identity) information is not available in our test sets, we use gender information as a proxy to study the effect of speaker changes. The notation '**f** → **m**' in Tables 1 and 2 denotes a test subset in which each target (incoming) utterance and its immediate previous one are from a male and female speaker, respectively (similar to the example presented in Section 1). The other notations can be interpreted similarly. To compute frame-level updates for the target utterance, $\boldsymbol{\mathcal{A}}|\boldsymbol{b}$ and $\boldsymbol{\mathcal{L}}|\boldsymbol{b}$ use the best DNN posterior as a substitute for alignments, while $\boldsymbol{\mathcal{A}}|\boldsymbol{p}$ and $\boldsymbol{\mathcal{L}}|\boldsymbol{p}$ use the top K (=10) DNN posteriors as a substitute for lattices. The following observations can be made from Tables 1 and 2.

• For segmental i-vectors, lattice posteriors offer little benefit over 1-best HMM alignments in general (columns 1 and 2). Note that $\boldsymbol{\mathcal{L}}|\cdot$ is slightly worse than $\boldsymbol{\mathcal{A}}|\cdot$ in most cases.

• For gender-matched cases (rows 1 and 2), no i-vector implementation offers a significant gain over $\boldsymbol{\mathcal{A}}|\cdot$, which is in line with our expectations.

• For gender-mismatched cases (rows 3 and 4), frame-level i-vectors are significantly better than $\boldsymbol{\mathcal{A}}|\cdot$, achieving up to 6.2% and 4.3% relative reduction in WER for *Test1* and *Test2*, respectively. Lattice posteriors appear to impart more robustness compared to 1-best alignments (columns 3 vs. 4; columns 5 vs. 6). Using the top K posteriors provides moderate gains over using the best posterior (columns 3 vs. 5; columns 4 vs. 6).

Table 1: *Relative WER reduction (%) over "$\boldsymbol{\mathcal{A}}|\cdot$", for different subsets of Test1. The highest value in each row is bold faced.*

	segmental		**frame-level**			
	$\boldsymbol{\mathcal{A}}\|\cdot$	$\boldsymbol{\mathcal{L}}\|\cdot$	$\boldsymbol{\mathcal{A}}\|\boldsymbol{b}$	$\boldsymbol{\mathcal{L}}\|\boldsymbol{b}$	$\boldsymbol{\mathcal{A}}\|\boldsymbol{p}$	$\boldsymbol{\mathcal{L}}\|\boldsymbol{p}$
1. **m** → **m**	0.0	-0.1	0.9	0.9	**1.1**	0.9
2. **f** → **f**	0.0	**0.3**	-0.2	0.0	0.2	0.2
3. **f** → **m**	0.0	-0.8	3.7	5.3	4.6	**6.2**
4. **m** → **f**	0.0	-2.3	0.1	**3.2**	0.7	2.1
5. **overall**	0.0	-0.1	0.9	1.1	1.1	**1.3**

Table 2: *Relative WER reduction (%) over "$\boldsymbol{\mathcal{A}}|\cdot$", for different subsets of Test2. The highest value in each row is bold faced.*

	segmental		**frame-level**			
	$\boldsymbol{\mathcal{A}}\|\cdot$	$\boldsymbol{\mathcal{L}}\|\cdot$	$\boldsymbol{\mathcal{A}}\|\boldsymbol{b}$	$\boldsymbol{\mathcal{L}}\|\boldsymbol{b}$	$\boldsymbol{\mathcal{A}}\|\boldsymbol{p}$	$\boldsymbol{\mathcal{L}}\|\boldsymbol{p}$
1. **m** → **m**	0.0	-0.4	**0.4**	-0.1	**0.4**	-0.1
2. **f** → **f**	0.0	-0.5	**0.5**	0.3	**0.5**	0.2
3. **f** → **m**	0.0	-1.5	2.7	**4.3**	3.9	**4.3**
4. **m** → **f**	0.0	-0.9	2.9	3.4	2.7	**3.6**
5. **overall**	0.0	-0.4	0.9	0.8	**1.1**	0.9

• Overall (row 5), frame-level updates seem to provide only marginal gains over segmental updates. This could be attributed to the fact that a large percentage ($>$ 75%) of our test utterances have a gender-matched context.

In essence, lattice posteriors are not useful in isolation but they are somewhat complementary to the information provided by frame-level updates. Frame-level updates are most effective in the presence of speaker (gender, in this case) changes, thereby corroborating the main hypothesis of this study. Lattice posteriors and frame-level updates show similar behaviors for *Test1* and *Test2*, but the effective adaptation gains are more significant for the latter considering that $\boldsymbol{\mathcal{A}}|\cdot$ provides a relative WER reduction of 7.1% over the LFBE-only model.

6. Conclusions

The main motivation of this paper was to investigate online i-vector implementations that can provide robust DNN adaptation in multi-speaker environments. To this end, the GMM-HMM i-vector framework of [1] was extended by: (a) computing frame-level i-vector updates based on the incoming target utterance; and (b) using posteriors from decoding lattices to accumulate statistics on the utterance history. A Bayesian view of i-vectors was also developed to formalize these extensions. Experiments with two test sets showed that: (a) lattice posteriors are somewhat complementary to frame-level updates but are not particularly helpful in isolation; (b) frame-level updates provide gains over segmental updates, and the gains are significant in the presence of speaker switches across utterances; and (c) adaptation gains are significantly higher when the microphone characteristics of test utterances are not seen in training.

7. Acknowledgments

The authors thank B. Hoffmeister, S. H. K. Parthasarathi and S. Matsoukas for their valuable feedback on the manuscript.

8. References

[1] S. Garimella, A. Mandal, N. Strom, B. Hoffmeister, S. Matsoukas, and S. H. K. Parthasarathi, "Robust i-vector based adaptation of DNN acoustic model for speech recognition," in *Proceedings of Interspeech*, 2015, pp. 2877–2881.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] P. Swietojanski, A. Ghoshal, and S. Renals, "Revisiting hybrid and GMM-HMM system combination techniques," in *Proceedings of ICASSP*, 2013, pp. 6744–6748.

[4] S. H. K. Parthasarathi, B. Hoffmeister, S. Matsoukas, A. Mandal, N. Strom, and S. Garimella, "fMLLR based feature-space speaker adaptation of DNN acoustic models," in *Proceedings of Interspeech*, 2015, pp. 3630–3634.

[5] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proceedings of ICASSP*, 2013, pp. 7947–7951.

[6] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proceedings of ICASSP*, 2013, pp. 7893–7897.

[7] P. Swietojanski, J. Li, and S. Renals, "Learning hidden unit contributions for unsupervised acoustic model adaptation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 8, pp. 1450–1463, 2016.

[8] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proceedings of ICASSP*, 2013, pp. 7398–7402.

[9] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proceedings of ASRU*, 2013, pp. 55–59.

[10] A. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with i-vector inputs," in *Proceedings of ICASSP*, 2014, pp. 225–229.

[11] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," in *Proceedings of Interspeech*, 2015, pp. 2440–2444.

[12] A. Ivanov, V. Ramanarayanan, D. Suendermann-Oeft, M. Lopez, K. Evanini, and J. Tao, "Automated speech recognition technology for dialogue interaction with non-native interlocutors," in *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2015, pp. 134–138.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *the IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[14] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[15] P. Kenny, "Joint factor analysis of speaker and session variability: Theory and algorithms," *CRIM, Montreal, (Report) CRIM-06/08-13*, 2005.

[16] S. Tibrewala and H. Hermansky, "Multi-band and adaptation approaches to robust speech recognition," in *Proceedings of Eurospeech*, 1997.

[17] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[18] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proceedings of Interspeech*, 2013, pp. 2345–2349.