Learning Hidden Unit Contributions for Unsupervised Acoustic Model Adaptation

Citation for published version:

Digital Object Identifier (DOI):
10.1109/TASLP.2016.2560534

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
IEEE/ACM Transactions on Audio, Speech, and Language Processing

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Learning Hidden Unit Contributions for Unsupervised Acoustic Model Adaptation

Pawel Swietojanski, Student Member, IEEE, Jinyu Li, Member, IEEE, and Steve Renals, Fellow, IEEE

Abstract—This work presents a broad study on the adaptation of neural network acoustic models by means of learning hidden unit contributions (LHUC) – a method that linearly re-combines hidden units in a speaker- or environment-dependent manner using small amounts of unsupervised adaptation data. We also extend LHUC to a speaker adaptive training (SAT) framework that leads to more adaptable DNN acoustic model, working both in a speaker-dependent and a speaker-independent manner, without the requirement to maintain auxiliary speaker-dependent feature extractors or to introduce significant speaker-dependent changes to the DNN structure. Through a series of experiments on four different speech recognition benchmarks (TED talks, Switchboard, AMI meetings and Aurora4) and over 270 test speakers we show that LHUC in both its test-only and SAT variants results in consistent word error rate reductions ranging from 5% to 23% relative depending on the task and the degree of mismatch between training and test data. In addition we have investigated the effect of the amount of adaptation data per speaker, the quality of unsupervised adaptation targets, the complementarity to other adaptation techniques, one-shot adaptation, and an extension to adapting DNNs trained in a sequence discriminative manner.

I. INTRODUCTION AND SUMMARY

SPEECH recognition accuracies have improved substantially over the past several years through the use of (deep) neural network (DNN) acoustic models. Hinton et al [1] report word error rate (WER) reductions between 10–32% across a wide variety of tasks, compared with discriminatively trained Gaussian mixture model (GMM) based systems. These results use neural networks as part of both hybrid DNN/HMM (hidden Markov model) systems [1]–[5] in which the neural network provides a scaled likelihood estimate to replace the GMM, and as tandem or bottleneck feature systems [6], [7] in which the neural network is used as a discriminative feature extractor for a GMM-based system. For many tasks it has been observed that GMM-based systems (with tandem or bottleneck features) that have been adapted to the talker are more accurate than unadapted hybrid DNN/HMM systems [8]–[10], indicating that the adaptation of DNN acoustic models is an important topic that merits investigation.

Acoustic model adaptation [11] aims to normalise the mismatch between training and runtime data distributions that arises owing to the acoustic variability across speakers, as well as other distortions introduced by the channel or acoustic environment. In this paper we investigate unsupervised model-based adaptation of DNN acoustic models to speakers and to acoustic environments, using a recently introduced method called Learning Hidden Unit Contributions (LHUC) [12]–[14]. We present the LHUC approach both in the context of test-only adaptation, and an extension to speaker-adaptive training (SAT), referred to as SAT–LHUC [14]. We present an extensive experimental analysis using four standard corpora: TED talks [15], AMI [16], Switchboard [17] and Aurora4 [18]. These experiments include: adaptation of both cross-entropy and sequence trained DNN acoustic models (Sec. VI-A–VI-C); an analysis in terms of the quality of adaptation targets, quality of adaptation data and the amount of adaptation data (Sec. VI-D); complementarity with feature-space adaptation techniques based on maximum likelihood linear regression [19] (Sec. VI-E); and application to combined speaker and environment adaptation (Sec. VII).

II. REVIEW OF NEURAL NETWORK ACOUSTIC ADAPTATION

Approaches to the adaptation of neural network acoustic models can be considered as operating either in the feature space, or in the model space, or as a hybrid approach in which speaker-, utterance-, or environment-dependent auxiliary features are appended to the standard acoustic features.

The dominant technique for estimating feature space transforms is constrained (feature-space) MLLR, referred to as CMLLR or fMLLR [19]. fMLLR is an adaptation method developed for GMM-based acoustic models, in which an affine transform of the input acoustic features is estimated by maximising the log-likelihood that the model generates the adaptation data based on first pass alignments. To use fMLLR with a DNN-based system, it is first necessary to train a complete GMM-based system, which is then used to estimate a single input transform per speaker. The transformed feature vectors are then used to train a DNN in a speaker adaptive manner and another set of transforms is estimated (using the GMM) during evaluation for unseen speakers. This technique has been shown to be effective in reducing WER across several different data sets, in both hybrid and tandem approaches [1], [4], [8], [9], [20]–[23]. Similar techniques have also been developed to operate directly on neural networks. The linear input network (LIN) [24], [25] defines an additional
speaker-dependent layer between the input features and the first hidden layer, and thus has a similar effect to fMLLR. This technique has been further developed to include the use of a tied variant of LIN in which each of the input frames is constrained to have the same linear transform \([4], [26]\). LIN and tied-LIN have been mostly used in test-only adaptation schemes; to make use of fMLLR transforms one needs to perform SAT training, which can usually better compensate against variability in acoustic space.

An alternative speaker-adaptive training approach – auxiliary features – augments the acoustic feature vectors with additional speaker-specific features computed for each speaker at both training and test stages. There has been considerable recent work exploring the use of i-vectors \([27]\) for this purpose. I-vectors, which can be regarded as basis vectors which span a subspace of speaker variability, were first used for adaptation in a GMM framework by Karafiat et al \([28]\). Saon et al \([29]\) used i-vectors to augment the input features of DNN-based acoustic models, and showed that appending 100-dimensional i-vectors for each speaker resulted in a 10\% relative reduction in WER on Switchboard (and a 6\% reduction when the input features had been transformed using fMLLR). Gupta et al \([30]\) obtained similar results, and Karanasou et al \([31]\) presented an approach in which the i-vectors were factorised into speaker and environment parts. Miao et al \([32]\) proposed to transform i-vectors using an auxiliary DNN which produced speaker-specific transforms of the original feature vectors, similar to fMLLR. Other examples of auxiliary features include the use of speaker-specific bottleneck features obtained from a speaker separation DNN used in a distant speech recognition task \([33]\), the use of out-of-domain tandem features \([23]\), and speaker codes \([34]–[36]\) in which a specific set of units for each speaker has been used to augment the input features of DNN-based acoustic models. The key idea of this approach is that the adaptation is performed by linear re-combination of the weights \([37]\) simply by splicing the models in \((1)\) \(m\) times. The properties also hold true when considering deeper (nested) models \([51]\) (Corollaries 2.6 and 2.7).

DNN training results in the hidden units learning a joint representation of the target function and becoming specialised and complementary to each other. Generalisation corresponds to the learned combination of basis functions continuing to approximate the target function when applied to unseen test data. This interpretation motivates the idea of using LHUC–Learning Hidden Unit Contributions – for test-set adaptation. In LHUC the network’s basis functions, previously estimated using a large amount of training data, are kept fixed. Adaptation involves modifying the combination of hidden units in order to minimise the adaptation loss based on the adaptation data. Fig. 1 illustrates this approach for a regression problem, where the adaptation is performed by linear re-combination of basis functions changing only the \(r\) parameters from eq. \((1)\).

In \((1)\) \(\psi : \mathbb{R} \rightarrow \mathbb{R}\) is an element-wise non-linear operation performed after an affine transformation which forms an adaptive basis function parameterised by a set of biases \(b_k \in \mathbb{R}\) and a weight vector \(w_k \in \mathbb{R}^{d_k}\). The target adaptation may then be constructed as a linear combination of the basis functions, each weighted by \(r_k \in \mathbb{R}\). The formulation can be extended to \(m\)-dimensional mappings \(f^{\mathcal{H}} : \mathbb{R}^d \rightarrow \mathbb{R}^m\) simply by splicing the models in \((1)\) \(m\) times. The properties also hold true when considering deeper (nested) models \([51]\) (Corollaries 2.6 and 2.7).

III. LEARNING HIDDEN UNIT CONTRIBUTIONS (LHUC)

A neural network may be viewed as a set of adaptive basis functions. Under certain assumptions on the family of target functions \(f^*\) (as well as on the model structure itself) the neural network can act as an universal approximator \([51]–[53]\). That is, given some vector of input random variables \(x \in \mathbb{R}^d\) there exists a neural network \(f_n(x) : \mathbb{R}^d \rightarrow \mathbb{R}\) of the form

\[
f_n(x) = \sum_{k=1}^{n} r_k \psi(w_k^T x + b_k)
\]

which can approximate \(f^*\) with an arbitrarily small error \(\epsilon\) with respect to a distance measure such as mean square error (provided \(n\) is sufficiently large):

\[
||f^*(x) - f_n(x)||_2 \leq \epsilon.
\]

In \((1)\) \(\psi : \mathbb{R} \rightarrow \mathbb{R}\) is an element-wise non-linear operation applied after an affine transformation which forms an adaptive basis function parameterised by a set of biases \(b_k \in \mathbb{R}\) and a weight vector \(w_k \in \mathbb{R}^{d_k}\). The target approximation may then be constructed as a linear combination of the basis functions, each weighted by \(r_k \in \mathbb{R}\). The formulation can be extended to \(m\)-dimensional mappings \(f^{\mathcal{H}} : \mathbb{R}^d \rightarrow \mathbb{R}^m\) simply by splicing the models in \((1)\) \(m\) times. The properties also hold true when considering deeper (nested) models \([51]\) (Corollaries 2.6 and 2.7).

DNN training results in the hidden units learning a joint representation of the target function and becoming specialised and complementary to each other. Generalisation corresponds to the learned combination of basis functions continuing to approximate the target function when applied to unseen test data. This interpretation motivates the idea of using LHUC–Learning Hidden Unit Contributions – for test-set adaptation. In LHUC the network’s basis functions, previously estimated using a large amount of training data, are kept fixed. Adaptation involves modifying the combination of hidden units in order to minimise the adaptation loss based on the adaptation data. Fig. 1 illustrates this approach for a regression problem, where the adaptation is performed by linear re-combination of basis functions changing only the \(r\) parameters from eq. \((1)\).

The key idea of LHUC is to explicitly parametrise the amplitudes of each hidden unit (either in fully-connected and convolutional layers after max-pooling), using a speaker-dependent amplitude function. Let \(h_j^{l,s}\) denote the \(j\)-th hidden unit activation (basis) in layer \(l\), and let \(r_j^{l,s} \in \mathbb{R}\) denote the \(s\)-th speaker-dependent amplitude function:

\[
h_j^{l,s} = \xi(r_j^{l,s}) \circ \psi_j \left( w_j^T x + b_j^s \right).
\]

where \(\xi(r)\) is an element-wise non-linear operation applied after an affine transformation which forms an adaptive basis function parameterised by a set of biases \(b_k \in \mathbb{R}\) and a weight vector \(w_k \in \mathbb{R}^{d_k}\). The target approximation may then be constructed as a linear combination of the basis functions, each weighted by \(r_k \in \mathbb{R}\). The formulation can be extended to \(m\)-dimensional mappings \(f^{\mathcal{H}} : \mathbb{R}^d \rightarrow \mathbb{R}^m\) simply by splicing the models in \((1)\) \(m\) times. The properties also hold true when considering deeper (nested) models \([51]\) (Corollaries 2.6 and 2.7).
The amplitude is modelled using a function $\xi : \mathbb{R} \rightarrow \mathbb{R}^+$ – typically a sigmoid with range $(0, 2)$ [13], but an identity function could be used [54]. $w_j$ is the $j$th column of the corresponding weight matrix $W$, $b_j$ denotes the bias, $\psi$ is the hidden unit activation function (unless stated otherwise, this is assumed to be sigmoid), and $\odot$ denotes a Hadamard product\(^1\). $\xi$ constrains the range of the hidden unit amplitude scaling (compare with Fig. 1) hence directly affecting the adaptation transform capacity – this may be desirable when adapting with potentially noisy unsupervised targets (see Sec. VI-A). LHUC adaptation progresses by setting the speaker-specific amplitude parameters $r_{j}$ using gradient descent with targets provided by the adaptation data.

The idea of directly learning hidden unit amplitudes was proposed in the context of an adaptive learning rate schedule by Trentin [55], and was later applied to supervised speaker adaptation by Abdel-Hamid and Jiang [12]. The approach was extended to unsupervised adaptation, non-sigmoid nonlinearities, and large vocabulary speech recognition by Swietojanski and Renals [13]. Other adaptive transfer function methods for speaker adaptation have also been proposed [44], [46], as have “basis” techniques [56]–[58]. However, the basis in the latter works involved re-tuning parallel models on pre-defined clusters (gender, speaker, environment) in a supervised manner; the adaptation then relied on learning linear combination coefficients for those sub-models on adaptation data.

\(^1\)Although the equations are given in scalar form, we have used Hadamard product notation to emphasise the operation that would be performed once expanded to full-rank matrices.

---

Fig. 1. Example illustration of how LHUC performs adaptation (best viewed in color). Top: A “bump” model (Eq. 1) with two hidden units can approximate “bump” functions. Middle: To learn function $f_2$ given training data $f_1$ (middle), we splice two “bump” functions together (4 hidden units, one input/output) to learn an approximation of function $f_2$. Bottom: LHUC adaptation of the model optimised to $f_2$ and adapted to $f_2$ using LHUC scaling parameters. Image reproduced from [14].

Fig. 2. A 4-hidden-unit model trained to $f_1(a)$ and $f_1(b)$ for an SI approach (top) and an adapted representation (middle) keeping the resulting basis functions fixed (bottom). (Best viewed in color.)

IV. SPEAKER ADAPTIVE TRAINING LHUC (SAT-LHUC)

When LHUC is applied as a test-only adaptation it assumes that the set of speaker-independent basis functions estimated on the training data provides a good starting point for further tuning to the underlying data distribution of the adaptation data (Fig. 1). However, one can derive a counter-example where this assumption fails: the top plot of Fig. 2 shows example training data uniformly drawn from two competing distributions $f_1(a)$ and $f_1(b)$ where the linear recombination of the resulting basis in the average model (Fig. 2 bottom), provides a poor approximation of adaptation data.

This motivates combining LHUC with speaker adaptive training (SAT) [59] in which the hidden units are trained to capture both good average representations and speaker-specific representations, by estimating speaker-specific hidden unit amplitudes for each training speaker. This is visualised in Fig. 3 where, given the prior knowledge of which data-point comes from which distribution, we estimate a set of parallel LHUC transforms (one per distribution) as well as one extra transform which is responsible for modelling average properties. The top of Fig. 3 shows the same experiment as in Fig 2 but with three LHUC transforms – one can see that the 4-hidden-unit MLP in this scenario was able to capture each of the underlying distributions as well as the average aspect well, given the LHUC transform. At the same time, the resulting basis functions (Fig 3, bottom) are a better starting point for the adaptation (Fig. 3, middle).

The examples presented in Figs. 2 and 3 could be solved by breaking the symmetry through rebalancing the number of training data-points for each function, resulting in less trivial and hence more adaptable basis functions in the average model. However, as we will show experimentally later, similar
The effects are also present in high-dimensional speech data, and SAT-LHUC training allows more tunable canonical acoustic models to be built, that can be better tailored to particular speakers through adaptation.

Test-only adaptation for SAT-LHUC remains the same as for LHUC— the set of speaker-dependent LHUC parameters \( \theta_{s}^{LHUC} = \{r^{l,s}\} \) is inserted for each test speaker and their values optimised from unsupervised adaptation data. We also use a set of LHUC transforms \( \Theta_{s}^{LHUC} \), where \( s = 1 \ldots S \), for the training speakers which are jointly optimised with the speaker-independent parameters \( \Theta_{sI} = \{W^{l}, b^{l}\} \). There is an additional speaker-independent LHUC transform, denoted by \( \theta_{0}^{LHUC} \), which allows the model to be used in speaker-independent fashion, for example, to produce first pass adaptation targets. This joint learning process of hidden units with speaker-dependent LHUC scalers is important, as it results in a more tunable canonical acoustic model that can be better adjusted to unseen speakers at test time, as we have illustrated in Fig. 3 and demonstrated on adaptation tasks in the following sections.

To perform SAT training with LHUC, we use the negative log likelihood and maximise the posterior probability of obtaining the correct context-dependent tied-state \( c_{t} \) given observation vector \( x_{t} \) at time \( t \):

\[
\mathcal{L}_{SAT}(\theta_{SI}, \theta_{SD}) = -\sum_{t \in D} \log P(c_{t}|x_{t}^{l}; \theta_{SI}; \theta_{0}^{LHUC})
\]  

where \( s \) denotes the \( s \)th speaker, \( m_{t} \in \{0, s\} \) selects the SI or SD LHUC transforms from \( \theta_{SD} \in \{\theta_{0}^{LHUC}, \ldots, \theta_{s}^{LHUC}\} \) based on a Bernoulli distribution:

\[
k_{t} \sim \text{Bernoulli}(\gamma)
\]

\[
m_{t} = \begin{cases} s & \text{if } k_{t} = 0 \\ 0 & \text{if } k_{t} = 1 \end{cases}
\]

where \( \gamma \) is a hyper-parameter specifying the probability the given example is treated as SI. The SI/SD split (determined by equations (5) and (6)) can be performed at speaker, utterance or frame level. We further investigate this aspect in section VI-B. The SAT-LHUC model structure is depicted in Fig 4; notice the alternative routes of forward and backward passes for different speakers.

Denote by \( \partial \mathcal{L}_{SAT}/\partial r^{l,s}_{j} \) the error back-propagated to the \( j \)th unit at the \( l \)th layer (eq. (3)). To back propagate through the transform one needs to element-wise multiply it by the transform itself, as follows:

\[
\frac{\partial \mathcal{L}_{SAT}}{\partial r^{l,s}_{j}} = \frac{\partial \mathcal{L}_{SAT}}{\partial h^{l,s}_{j}} \circ \xi(r^{l,s}_{j}).
\]

To obtain the gradient with respect to \( r^{l,s}_{j} \):

\[
\frac{\partial \mathcal{L}_{SAT}}{\partial r^{l,s}_{j}} = \frac{\partial \mathcal{L}_{SAT}}{\partial h^{l,s}_{j}} \circ \xi(r^{l,s}_{j}) \circ \psi_{j}.
\]

When performing mini-batch SAT training one needs to explicitly take account of the fact that different data-points may flow through different transforms: hence the resulting gradient for \( r^{l,s}_{j} \) for the \( s \)th speaker is the sum of the partial gradients belonging to speaker \( s \):

\[
\frac{\partial \mathcal{L}_{SAT}}{\partial r^{l,s}_{j}} = \sum_{t, m_{t} = s} \frac{\partial \mathcal{L}_{SAT}}{\partial h^{l,s}_{j}} \circ \frac{\partial \xi(r^{l,s}_{j})}{\partial r^{l,s}_{j}} \circ \psi_{j}.
\]
V. EXPERIMENTAL SETUPS

We experimentally investigated LHUC and SAT–LHUC using four different corpora: the TED talks corpus [15] following the IWSLT evaluation protocol (www.iwslt.org); the Switchboard corpus of conversational telephone speech [17] (lrc.upenn.edu); the AMI meetings corpus [16], [60] (corpus amiiproject.org); and the Aurora4 corpus of read speech with artificially corrupted acoustic environments [18] (catalog.elra.info). Unless explicitly stated otherwise, the models share similar structure across the tasks – DNNs with 6 hidden layers, 2,048 units in each layer using 40 mel-filter-bank (FBANK) features for which fMLLR transforms cannot be easily obtained (though not impossible [64]), and for which LHUC offers an interesting adaptation alternative. We also evaluated the effectiveness of LHUC and SAT–LHUC applied to convolutional networks [8], [65], [66], trained with 300 convolutional filters. We decoded with a pruned 3-gram language model estimated from 800k words of AMI training transcripts interpolated with an LM trained on Fisher conversational telephone speech transcripts (1M words).

Switchboard: We use the Kaldi GMM recipe [68], [69], using Switchboard–1 Release 2 (LDC97S62). Our baseline unadapted acoustic models were trained on LDA/MLLT features. The results are reported on the full Hub5 00 set (LDC2002S09) to which we will refer as eval2000. The eval2000 contains two types of data, Switchboard (SWBD) – which is better matched to the training data – and CallHome English (CHE). Our reported results use 3-gram LMs estimated from Switchboard and Fisher data.

Aurora4: The Aurora 4 task is a small scale, medium vocabulary noise and channel ASR robustness task based on the Wall Street Journal corpus [18]. We train our ASR models using the multi-condition training set. One half of the training utterances were recorded using a primary Sennheiser microphone, and the other half was collected using one of 18 other secondary microphones. The multi-condition set contains noisy utterances corrupted with one of six different noise types (airport, babble, car, restaurant, street traffic and train station) at 10-20 dB SNR. The standard Aurora 4 test set (eval192) consists of 330 utterances, which are used in 14 test conditions (4620 utterances in total). The same six noise types used during training are used to create noisy test utterances with SNRs ranging from 5-15dB SNR, resulting in a total of 14 test sets. These test sets are commonly grouped into 4 subsets – clean (group A, 1 test case), noisy (group B, 6 test cases), clean with channel distortion (group C, 1 test case) and noisy with channel distortion (group D, 6 test cases). We decode with the standard task’s 5k words bigram LM.

VI. RESULTS

A. LHUC hyperparameters

Our initial study concerned the hyper-parameters used with LHUC adaptation. First, we used the TED talks to investigate how the word error rate (WER) is affected by adapting different layers in the model using LHUC transforms. The results, graphed in Fig. 5 (a), indicated that adapting only the bottom layer brings the largest drop in WER; however, adapting more layers further improves the accuracy for both LHUC and SAT–LHUC approaches (adapting the other way round – starting from the top layer – is much less effective [13]). Since obtaining the gradients for the r parameters at each layer is inexpensive compared to the overall back-propagation, and we want to adapt at least the bottom layer, we apply LHUC to each layer for the rest of this work.

Fig. 5 (b) shows WERs for the number of adaptation iterations. The results indicate that one sweep over the adaptation data (in this case tst2010) is sufficient and, more importantly, the model does not overfit when adapting with more iterations (despite the adaptation objective consistently
improving – Fig. 5 (c)). This suggests that it is not necessary to carefully regularise the model – for example, by Kullback-Leibler divergence training [38] which is usually required when adapting the weights of one or more layers in a network.

Finally, we explored how the form of the LHUC re-parametrisation function \( \xi \) affects the WER and frame error rate (FER) (Fig. 5 (c) and Table II). For test-only adaptation only a small WER difference (0.1% absolute) is observed, regardless of the large difference in frame accuracies. This supports our previous observation that LHUC is robust against over-fitting. For SAT-LHUC training, a less constrained parameterisation was found to give better WERs for the SI model. Based on our control experiments, during SAT-LHUC training, setting \( \xi \) to be the identity function (linear \( r \)) gave similar results to \( \xi(r) = \max(0, r) \) and \( \xi(r) = \exp(r) \) and all were better than re-parametrisating with \( \xi(r) = 2/(1 + \exp(-r)) \). This is expected as for full training the last approach constrains the range of back-propagated gradients. From now on, if not stated otherwise, we will use \( \xi(r) = \exp(r) \) in the remainder of this paper.

We adapt our all models with the learning rate set to 0.8 (regardless of \( \gamma \)) and the basic training of both the SI and the SAT-LHUC models was performed with the initial learning rate set to 0.08 and was later adjusted according to the newbob learning scheme [70].

B. SAT-LHUC

As described in section IV, SAT-LHUC training aims to regularise the hidden unit feature receptors so that they capture not just the average characteristics of training data, but also specific features of the different distributions the data was drawn from (for example, different training speakers). As a result, the model can be better tailored to unseen speakers by putting more importance to those units that were useful for training speakers with similar characteristics.

Prior to SAT-LHUC training we need to decide on how and which data should be used to estimate speaker-dependent and speaker-independent transforms. In this work we train SAT-LHUC models with frame-level [14], segment-level and speaker-level clusters. For speaker- and segment-level transforms we decide which speakers or segments are going to be treated as SI or SD prior to training. For the frame-level SAT-LHUC approach, the SI/SD decisions are made separately for each data-point during training. In either scenario we ensure that the overall SD/SI ratio determined by \( \gamma \) parameter is satisfied. The WER results for each of these three approaches (\( \gamma = 0.5 \)) are reported in Table III. Speaker-level SAT-LHUC training provides the highest WERs for both SI and SD decodes. Segment-level and frame-level SAT-LHUC training result in similar WERs for SI decodes, with a small advantage (0.1% abs.) for the frame-level approach after adaptation.

![Fig. 6. WER(%) for different sampling strategies]{per frame, per segment, per speaker) for SAT-LHUC training and SI and SD decodes on TED tst2013.}

We report the baseline LHUC and SAT-LHUC comparisons on TED and AMI data in Tables IV and V, respectively.
further results, including a comparison to fMLLR transforms and on Switchboard data are in the next sections). On TED (Table IV), SAT-LHUC models operating in SI mode ($\gamma = 0.6$) have comparable WERs to SI models; however, adaptation resulted in a WER reduction of 0.3–1.1% absolute (2–6% relative) compared to test-only adaptation of the SI models. Similar results were observed on the AMI data (Table V) where for both DNN and CNN models trained on FBANK features LHUC adaptation decreased the WER by 2% absolute (7% relative) and SAT-LHUC training improved this result by 4% relative for DNN models. As expected, the SAT-LHUC gain for CNNs was smaller when compared to DNN models, since the CNN layer can learn different patterns for different speakers which may be selected through the max-pooling operator at run-time.

C. Sequence model adaptation

Model-based adaptation of sequence-trained DNNs (SE-DNN) is more challenging compared to adapting networks trained using cross-entropy: a mismatched adaptation objective (here cross-entropy) can easily erase sequence information from the weight matrices due to the well-known effect of catastrophic forgetting [71] in neural networks. Indeed Huang and Gong [39] report no gain from adapting SE-DNN models with cross-entropy adaptation objective and supervised adaptation targets. In those experiments, all weights in the model were updated and one needs to perform KL divergence regularised adaptation [38] or KL regularised sequence level adaptation to further improve on top of SE-DNN. It remains to be answered if one can get similar improvements using SE-DNN adaptation and first-pass transcripts.

In this work we adapt state-level minimum Bayes risk (sMBR) [72], [73] sequence-trained models using LHUC and report results on TED tst2011 and tst2013 in Table VI. We kept all the LHUC adaptation hyper-parameters the same as for CE models and obtained around 2% absolute (11% relative) WER reductions on tst2013 for both SI and fMLLR SAT adapted SE-DNN systems. Interestingly, the obtained adaptation gain was similar to the cross-entropy models and LHUC adaptation did not seem to disrupt the learned model’s sequence representation.

We compared our adaptation results to the most accurate system of the IWSLT–2013 TED transcription evaluation, which performed both feature- and model-space speaker adaptation [74]. For model-space adaptation that system used a method which adapts DNNs with a speaker-dependent layer [42]. The results are reported in Table VII where in the first block one can see a standard sequence-trained feature-space adapted system build from TED and 150 hours of out-of-domain data scoring 15.7% WER, similar to the WER of our TED system (15.4%), which also for IWSLT utilised 100 hours of out-of-domain AMI data. The 0.3% difference could
be explained by characteristics of the out-of-domain data used (tst2013 is characterised by a large proportion of non-native speakers which is also typical for AMI data, hence benefits more our baseline systems). When comparing both adaptation approaches operating in an unsupervised manner one can see that LHUC gives much bigger improvements in WER compared to speaker-dependent layer, 2.1\% vs. 0.6\% absolute (14\% vs. 4\% relative) on tst2013. This allows our single-model system to match a considerably more sophisticated post-processing pipeline [74], as outlined in Table VII. For less mismatched data (tst2011) adaptation is less important and our system has a WER 0.8\% absolute higher compared with the more sophisticated system.

From these experiments we conclude that LHUC is an effective way to adapt sequence models in an unsupervised manner using a cross-entropy objective function, without the risk of removing learned sequence information.

D. Other aspects of adaptation

Amount of adaptation data: Fig 7 shows the effect of the amount of adaptation data on WER for LHUC and SAT-LHUC adapted models. As little as 10s of unsupervised adaptation data is already able to substantially decrease WERs (by 0.5–0.8\% absolute). The improvement for SAT-LHUC adaptation compared with LHUC is considerably larger – roughly by a factor of two up to 30s adaptation data. As the duration of adaptation data increases the difference gets smaller; however SAT-LHUC results in consistently lower WERs than LHUC in all cases (including full two pass adaptation).

We also investigated supervised (oracle) adaptation by aligning the acoustics with the reference transcriptions (dashed lines). Given supervised adaptation targets, LHUC and SAT-LHUC further substantially decrease WERs, with SAT-LHUC giving a consistent advantage over LHUC.

Quality of adaptation targets: Since our approach relies on a first-pass decoding, we investigated the extent to which LHUC is sensitive to the quality of the adaptation targets. In this experiment we explored the differences resulting from different language models, and assumed that the first pass adaptation data was generated by either an SI or a SAT-LHUC model operating in SI mode. The main results are shown in Fig 8 where the solid lines show WERs obtained with a pruned 3-gram LM and different types of adaptation targets resulting from re-scoring the adaptation data with stronger LMs. One can see there is not much difference unless the adaptation data was re-scored with the largest 4-gram LM. This improvement diminishes in the final adapted system after re-scoring. This suggests that the technique is not very sensitive to the quality of adaptation targets. This trend holds regardless of the amount of data used for adaptation (ranging from 10s to several minutes per speaker). In related work [32] LHUC was employed using alignments obtained from an SI-GMM system with a 8.1\% absolute higher WER than the corresponding SI DNN, and substantial gains were obtained over the unadapted SI DNN baseline – although the WER reduction was considerably smaller (1\% absolute) compared to adaptation with alignments obtained with the corresponding SI DNN.

Quality of data: We also investigated how the quality of the acoustic data itself affects the adaptation accuracies, keeping the other ASR components fixed. We performed an experiment on the AMI corpus using speech captured by individual headset microphones (IHM) and a single distant tabletop microphone (SDM). In case of IHM we adapt to the headset; in this experiment we assume we have speaker labels for the SDM data\textsuperscript{2}. The results are reported in Table VIII: LHUC adaptation improves the accuracy in both experiments, although the gain for the SDM condition is smaller; however, the SDM system is characterised by twice as large WERs. Notice that LHUC has also been successfully applied to channel normalisation between distant and close talking microphones [75].

One-shot adaptation: By one-shot adaptation we mean the scenario in which LHUC transforms were estimated once for a held-out speaker and then used many times in a single pass

\textsuperscript{2}In a real scenario for SDM data one would have to perform speaker diarisation in order to obtain speaker labels.
system for this speaker. We performed those experiments on AMI IHM data, and report results on dev and eval which contain 21 and 16 unique speakers taking part in 18 and 16 different meetings, respectively. Each speaker participates in multiple meetings: some degree, adapting to a speaker in one meeting, then applying the adaptation transform to the same speaker in the other meetings simulates a real-life condition where it is possible to assume the speaker identity without having to perform speaker diarisation (e.g. personal devices). The results of this experiment (Table IX) indicate that LHUC retains the accuracies of two-pass systems by providing almost identical results when comparing LHUC estimated in one meeting, then applying the adaptation transform to the same speaker in the other meetings simulates a real-life condition where it is possible to assume the speaker identity without having to perform speaker diarisation (e.g. personal devices). The results of this experiment (Table IX) indicate that LHUC retains the accuracies of two-pass systems by providing almost identical results when comparing LHUC estimated in a full two-pass system and when the unsupervised transforms are re-used in the LHUC one-shot experiment.

E. Complementarity to feature normalisation

Feature-space adaptation using fMLLR is a very reliable current form of speaker adaptation, so it is of great interest to explore how complementary the proposed approaches are to SAT training with fMLLR transforms.\(^3\)

We compared LHUC and SAT−LHUC to SAT-fMLLR training using TED tst2010 (Fig 9, red curves). We also compared both techniques, including a comparison in terms of the amount of data used to estimate each type of transform.

\(^3\)Due to space constraints we do not make an explicit comparisons to other techniques such as auxiliary i-vector features or speaker-codes; however, the literature suggest that the use of i-vectors give similar results when compared to fMLLR trained models. Related recent studies also show LHUC is at least as good as the standard use of i-vector features [32], [76].

fMLLR transforms estimated on 10s of unsupervised data result in an increase in WER compared with the SI-trained baseline (16.1% vs. 15.0%). When combined with LHUC or SAT−LHUC some of this deterioration was recovered (similar results using LHUC alone were reported in Fig 7). For more adaptation data (30s or more) fMLLR improved the accuracies by around 1–2% absolute and combination with LHUC (or SAT−LHUC) resulted in an additional 1% reduction in WER (see also Table X in the next section for further results).

We also investigated (in a rather unrealistic experiment) how much mismatch in feature space one can normalise in model space with LHUC. To do so, we used a SAT-fMLLR trained model with unadapted PLP features which gave a large increase in WER (26% vs 15%). Then, using unsupervised adaptation targets obtained from the feature-mismatched decoding both LHUC and SAT−LHUC were applied. The results (also presented in Fig. 9) indicate that a very large portion of the WER increase can be effectively compensated in model space – more than 8% absolute. As found before, test-only renormalisation functions (exp(r) vs. 2/(1 + exp(−r))) have negligible impact on the adaptation results, and SAT−LHUC again provides better results.

F. Adaptation Summary

In this section we summarise our results, applying LHUC and SAT−LHUC to TED, AMI, and Switchboard. Table X contains results for four IWSLT test sets (dev2010, tst2010, tst2011, and tst2013): in most scenarios SAT−LHUC
results in a lower WER than LHUC and both techniques are complementary with SAT-fMLLR training.

Similar conclusions can be drawn from experiments on AMI (Table XI) where LHUC and SAT-LHUC were found to effectively adapt DNN and CNN models trained on FBANK features. SAT-LHUC trained DNN models gave the same final results as the more complicated SAT-fMLLR+LHUC system.

On Switchboard, in contrast to other corpora, we observed that test-only LHUC does not match the WERs obtained from SAT-fMLLR models (Table XII). The SI system has a WER of 21.7% compared with 20.7% for the test-only LHUC and 20.2% for the SAT-fMLLR system. The improvement obtained using test-only LHUC is comparable to that obtained with other test-only adaptation techniques, e.g. feature-space discriminative linear regression (fDLR) [4], but neither of these matches SAT trained feature transform models. This could be due to the fact Switchboard data is narrow-band and as such contains less information for discrimination between speakers [77], especially when estimating relevant statistics from small amounts of unsupervised adaptation data. Another potential reason could be related to the fact that the Switchboard part of eval2000 is characterised by a large overlap between training and test speakers – 36 out of 40 test speakers are observed in training [78], which limits the need for adaptation, but also enables models to learn much more accurate speaker-characteristics during supervised speaker adaptive training.

Adaptation using SAT-LHUC (20.3% WER) almost matches SAT-fMLLR (20.2%). We also observe that LHUC performs relatively better under more mismatched conditions (the Callhome (CHE) subset of eval2000), similar to what we observed on TED.

Finally, in Fig 10 we show the WERs obtained for 200 speakers across the TED, AMI, and SWBD test sets. We observe that for 89% of speakers LHUC and SAT-LHUC adaptation reduced the WER, and that SAT-LHUC gives a consistent reduction over LHUC.

<table>
<thead>
<tr>
<th>TABLE X</th>
<th>WER (%) on various TED development and test sets from IWSLT12 and IWSLT13 evaluations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>dev2010</td>
</tr>
<tr>
<td>DNN</td>
<td>14.0</td>
</tr>
<tr>
<td>+LHUC</td>
<td>14.5</td>
</tr>
<tr>
<td>+SAT-LHUC</td>
<td>15.4</td>
</tr>
<tr>
<td>+fMLLR</td>
<td>14.3</td>
</tr>
<tr>
<td>++LHUC</td>
<td>14.1</td>
</tr>
<tr>
<td>++SAT-LHUC</td>
<td>13.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE XI</th>
<th>WER (%) on AMI–HMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Features</td>
</tr>
<tr>
<td>DNN</td>
<td>FMLLR</td>
</tr>
<tr>
<td>+LHUC</td>
<td>FMLLR</td>
</tr>
<tr>
<td>DNN</td>
<td>FBANK</td>
</tr>
<tr>
<td>+LHUC</td>
<td>FBANK</td>
</tr>
<tr>
<td>+SAT-LHUC</td>
<td>FBANK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE XII</th>
<th>WER (%) on Switchboard eval2000.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>SWB</td>
</tr>
<tr>
<td>DNN</td>
<td>15.2</td>
</tr>
<tr>
<td>+LHUC</td>
<td>14.7</td>
</tr>
<tr>
<td>++SAT-LHUC</td>
<td>14.6</td>
</tr>
<tr>
<td>+fMLLR</td>
<td>14.2</td>
</tr>
<tr>
<td>++LHUC</td>
<td>14.2</td>
</tr>
<tr>
<td>++SAT-LHUC</td>
<td>14.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE XIII</th>
<th>Results on Aurora 4. Multi-condition DNN model.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>A</td>
</tr>
<tr>
<td>DNN</td>
<td>5.1</td>
</tr>
<tr>
<td>DNN + rS</td>
<td>4.3</td>
</tr>
<tr>
<td>DNN + rE</td>
<td>5.0</td>
</tr>
<tr>
<td>DNN + rSE JOINT</td>
<td>4.5</td>
</tr>
<tr>
<td>DNN + rSE, α = 0.5</td>
<td>4.6</td>
</tr>
<tr>
<td>DNN + rSE, α = 0.7</td>
<td>4.5</td>
</tr>
</tbody>
</table>

VII. LHUC FOR FACTORIZATION

We applied LHUC to adapt to both the speaker and the acoustic environment. If multi-condition data is available for a speaker, then it is possible to define a set of joint speaker-environment LHUC transforms. Alternatively, we can estimate two sets of transforms – for speaker rS and for environment rE – and then linearly interpolate them, using hyper-parameter α, to derive a combined transform řSE as follows:

\[ \xi (\hat{r}^{S}_{SE}) = \alpha \xi (r^{S}_{E}) + (1-\alpha) \xi (r^{E}_{E}) \]  

(10)

Notice, that although both types of transforms are estimated in an unsupervised manner we assume that the test environment is known, allowing the correct environmental transform to be selected. This adaptation to the test environment is similar to that of Li et al [79].

We adapted baseline multi-condition trained DNN models [80] to the speaker (rS) and the environment (rE). The rS transforms were estimated only on clean speech; similarly the environment transforms were estimated for each scenario (one set of rE per scenario) using multiple speakers (hence, we have 7 different environmental transforms). To avoid learning joint speaker-environment transforms the target speaker’s data was removed from environment adaptation material (e.g. when estimating transforms for the restaurant environment, we use all restaurant data except the one for the target speaker).

The results (Table XIII) show that both standalone speaker or environment adaptation LHUC adaptation improve over an unabated system (13.1% (S) and 13.3% (E) vs. 13.9%) but, as expected, a single transform estimated jointly on the target speaker and environment has a lower WER (12.4%). However, when interpolated with α = 0.7 the result of the factorised model improves to 12.7% WER, although still higher that joint estimation. However, adaptation data for joint speaker-environment adaptation is not available in many scenarios, and the factorised adaptation based on interpolation is more flexible.
Fig. 10. Summary of WERs(%) obtained with LHUC and SAT-LHUC adaptation techniques on test speakers of TED, SWBD and AMI corpora (results are sorted in descending WER order for the SI system). For LHUC the average observed improvement per speaker was at 1.6% absolute (7.0% relative). The same statistic for SAT-LHUC was at 2.3% absolute (9.7% relative). The maximum observed WER decrease per speaker was 11.4% absolute (32.7% relative) and 16.0% absolute (50% relative) for LHUC and SAT-LHUC, respectively. WERs decreased for 89% of speakers using LHUC adaptation.

TABLE XIV
RESULTS ON AURORA 4. MULTI-CONDITION MAXOUT-CNN MODEL, WITH AND WITHOUT ANNEALED DROPOUT (AD).

<table>
<thead>
<tr>
<th>Model</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxCNN</td>
<td>4.2</td>
<td>7.7</td>
<td>7.9</td>
<td>17.4</td>
<td>11.6</td>
</tr>
<tr>
<td>MaxCNN + rSE JOINT</td>
<td>3.7</td>
<td>6.3</td>
<td>5.5</td>
<td>14.3</td>
<td>9.5</td>
</tr>
<tr>
<td>AD MaxCNN</td>
<td>4.3</td>
<td>7.7</td>
<td>7.2</td>
<td>15.6</td>
<td>10.9</td>
</tr>
<tr>
<td>AD MaxCNN + rSE JOINT</td>
<td>3.4</td>
<td>5.7</td>
<td>6.1</td>
<td>13.4</td>
<td>8.7</td>
</tr>
</tbody>
</table>

We also trained more competitive models following Rennie et al [81]: Maxout [82] CNN models were trained using annealed dropout [83]. In this work we used alignments obtained by aligning a corresponding multi-condition model as ground-truth labels, rather than replicating clean alignments to multi-condition data, in contrast to [81]; this is likely to explain differences in the reported baselines (10.9% compared with 10.5% in [81]). The results for the joint optimisation are reported in Table XIV where one can notice large improvements with unsupervised LHUC adaptation.

Finally, we visualise the top hidden layer activations of the annealed dropout Maxout CNN using stochastic neighbourhood embedding (tSNE) [84] for one utterance recorded under clean and noisy (restaurant) conditions (Fig. 11).

VIII. CONCLUSIONS

We have presented the LHUC approach to unsupervised adaptation of neural network acoustic models in both test-only (LHUC) and SAT (SAT-LHUC) frameworks, evaluating them using four standard speech recognition corpora: TED talks as used in the IWSLT evaluations, AMI, Switchboard, and Aurora4. Our experimental results indicate that both LHUC and SAT-LHUC can provide significant improvements in word error rates (5–23% relative depending on test set and task). LHUC adaptation works well unsupervised and with small amounts of data (as little as 10s), is complementary to
feature space normalisation transforms such as SAT-fMLLR, and can be used for unsupervised adaptation of sequence-trained DNNS acoustic models using a cross-entropy adaptation objective function. Furthermore we have demonstrated that it can be applied in a factorised way, estimating and interpolating separate transforms for adaptation to the acoustic environment and speaker.

REFERENCES


