# NON-NEGATIVE INTERMEDIATE-LAYER DNN ADAPTATION FOR A 10-KB SPEAKER ADAPTATION PROFILE

*Kshitiz Kumar, Chaojun Liu, Yifan Gong*

Microsoft Corporation, Redmond, WA

{kshitiz.kumar, chaojunl, yifan.gong}@microsoft.com

## ABSTRACT

Previously we demonstrated that speaker adaptation of acoustic models (AM) can provide significant improvement in the accuracy of large-scale speech recognition systems. In this work we discuss numerous challenges in scaling speaker adaptation to millions of speakers, where the size of speaker-dependent (SD) parameters is a critical challenge. Subsequently, we formulate an intermediate-layer adaptation framework for adaptation, upon which we build a non-negative adaptation for a very sparse set of non-negative SD parameters. We further improve this work with, (a) non-negative adaptation with a small-positive threshold, (b) setting small-positive weights in an already trained non-negative model to zero. We also discuss effective methods to store the non-negative SD parameters. We show that our methods reduce the SD parameters from 86KB for our previous best adaptation approach to 8.8KB, thus about 90% relative reduction in the size of SD parameters, and still retain 10+% word-error-rate-relative (WERR) gain over the baseline speaker-independent (SI) model.

***Index Terms—*** Non-negativity, DNN, Speaker Adaptation, Digital Assistant, Personalization

## 1. INTRODUCTION

Our recent work [1] leveraged recent advances in deep neural networks (DNNs) and demonstrated that AM personalization using unsupervised adaptation techniques can provide 10+% word-error-rate-relative (WERR) gain for large scale speech recognition systems. Although recent advances in the strong modeling capability of DNNs [2, 3] have significantly improved the performance of SI models, adaptation techniques still provide a strong additional value to automatic speech recognition (ASR) customers. Similarly we expect adaptation to provide strong gains on top of the next generation deep learning techniques and thus continue to be an active research and development topic.

The adaptation approaches for DNNs can be classified in the following broad areas, (1) affine transformation - [4] applies affine transformation on the top layer activations, (2) adapting multiple layers - [5] leverages singular value decomposition (SVD) to achieve low-footprint adaptation, [6] learns speaker-specific hidden contributions, (3) subspace approaches - we construct an adapted DNN from analysis on different subspace models, see examples in principal component analysis [7], i-vectors [8, 9], and GMM fMLLR based approach in [10], (4) speaker adaptive training (SAT) in [11]. Most of the adaptation approaches leverage conservative training methods, where we constrain SD parameters to be in close vicinity of the SI model [12][13].

In this work in Sec. 2, we discuss a number of benefits and new challenges in successfully deploying speaker adaptation for millions of speakers. Some of the challenges arise from, (a) size of the SD parameters, (b) limited adaptation data, (c) unsupervised nature of data - we use hypotheses from decoding against SI model as approximate transcriptions, (d) potential regression for shared devices, (e) discarding corrupted or noisy data, (f) effectively testing models, (g) model adaptation throughput, (h) accuracy gains. The size of SD parameters is a critical challenge; a smaller model assists almost all of above challenges, and thus a major focus of this work.

In Sec. 3, we present an intermediate-layer adaptation framework. This is motivated to impact both the seen and unseen senones in the adaptation data, and also provides a small set of the SD parameters. Later in Sec. 4, we build a non-negative [14, 15] adaptation approach to achieve a very sparse set of SD parameters. We further improve this work with, (a) non-negative adaptation with a small-positive threshold, (b) setting small-positive weights in an already trained non-negative model to zero. We provide results in terms of, (a) effective non-zero percent of SD parameters, (b) impact on WER, and show that our approach yields a substantially smaller SD model. Next, we discuss three methods to effectively store the non-negative SD parameters. Finally we show that our methods reduce the SD parameters size from 86KB for our previous best adaptation approach to 8.8KB. This is an order of magnitude reduction in the SD parameters while still nearly retaining the WERR gain over the SI model. Sec. 5 concludes this study.

## 2. MOTIVATION AND CHALLENGES FOR PERSONALIZATION

Fig. 1 provides a brief outline of a typical speech recognition system. There the key modeling components are, (1) acoustic model (AM) represents a map between acoustic features and speech states, (2) pronunciation model represents words into speech states, (3) language model statistically models word sequences, (4) search engine. The goal of a large-scale adaptation system is to leverage one or more of these adaptation opportunities. In this work we focus on improvements from personalization of AM.

The benefits from speaker adaptation have been well established in literature. We verified that the relative gain from adaptation holds for newer deep learning techniques [1], so we expect adaptation to provide strong gains on top of the future modeling technologies. In general, the personalization techniques can be applied to, (a) a cluster of speakers or an individual speaker, (b) a cluster of devices or a particular device, (c) age or gender categories, (d) noise, reverberation etc. Personalization can account for different accents, speaking rate, and acoustic environment etc., and may also be critical to retain customers who are currently experiencing very high WER.

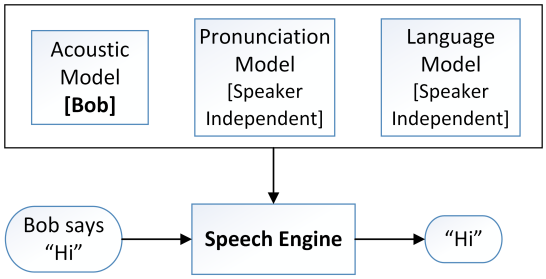


**Fig. 1**. *Personalization of acoustic model.*

### 2.1. Challenges to Personalization

Prior studies have focused on techniques to improve adaptation accuracy. However, we have seen limited treatment on key challenges in deploying speaker adaptation to millions of speakers. This work contributes towards better understanding the benefits and challenges in a large-scale speaker adaptation framework.

The challenges to personalization stem from a huge number of the SD models that need to built. In contrast to a single SI model, personalization requires building millions of SD models. This puts practical constraints on the number of SD parameters. A closely related issue is the quality of SD models. The SI model training has access to a rich set of transcribed data but a majority of adaptation work feeds on untranscribed data, where we use hypotheses from decoding against SI model as approximate transcriptions. In this context, it becomes important to seek ways to improve untranscribed data quality by leveraging confidence-scores [16], clicked-queries [17], completed-tasks *e.g.*, setting up calendar event etc. Limited adaptation data is another challenge for a large majority of speakers. We may also have challenges from shared devices, where adaptation on data from multiple users, sharing a device, may lead to regression for some [18]. Furthermore, adaptation must also meet the constraints of computation and latency. We demand a high throughput, in terms of no. of models trained per day. The design of adaptation must also address issues from potential future updates to the SI model. We believe that above challenges will motivate researchers in the field to develop effective solutions, and lead towards a wider push in deploying speaker adaptation for large user bases.

## 3. INTERMEDIATE-LAYER ADAPTATION FRAMEWORK

In this section we briefly outline our framework of intermediate-layer adaptation we presented in [1]. We discuss our framework with respect to a representative DNN model in Fig. 2(a). In practice our DNNs consist of 5 hidden layers but for current description, we consider a DNN with 2 non-linear hidden layers in (D0, D1) and an output layer. Our baseline DNN architecture also includes SVD layers (S0, S1) [5]. The non-linear hidden, linear SVD and output layers consist of 2048, 208 and 6k activation units, respectively. We use the notation "L-D0" to indicate DNN layer weights that input to D0, from which we compute output of layer D0, similarly L-S0 and L-O respectively indicates layers that input to S0 and *Output*.

Some of the recent work proposed adaptation on the bottom hidden layer (L-D0) or top layer (L-O) [4]. In [1] we demonstrated that we can simultaneously benefit both accuracy and SD parameters by adapting one of the intermediate layers. Adapting L-O only impacts the senones [19] seen in adaptation data - in this context the method exhibits similarity with MAP [20] of GMMs, where MAP requires

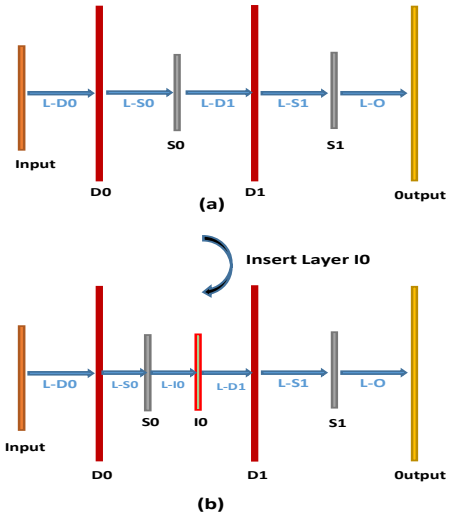


**Fig. 2**. *(a) A representative SVD-based DNN architecture; layers in (D0, D1) indicate usual deep non-linear layers; layers in (S0, S1) indicate linear SVD layers, (b) Inserting a compact linear layer I0 on top of SVD layer S0.*

sufficient observation on all senones. In contrast, an intermediate-layer adaptation exhibits similarities with transformation-based adaptations *e.g.* MLLR [21][22], where for limited data we can impact both seen and unseen senones due to inherent transformation through subsequent DNN layers. Correspondingly intermediate-layer adaptation requires far fewer parameters than that for top-layer adaptation. We note typical no. of parameters for a production-level DNN in Table 1, where we note that adapting intermediate-layers offers huge savings. Furthermore for ASR, we rationalize first few DNN layers as feature normalization steps, where device and speaker-dependent features get normalized; we think of middle-layers as higher-order feature synthesis, where we encapsulate normalized features into abstract speech bases; and finally top-layer is a classification layer that classifies speech into physical triphones or senone states. Thus individually adapting different intermediate DNN layers provides unique adaptation techniques.

### 3.1. Insert and adapt a linear layer on top of SVD layer

We can further improve the intermediate-layer adaptation framework with inserting and adapting a linear layer on top of an intermediate SVD layer [1]. We demonstrate the step of inserting a layer in Fig. 2(b); there we insert a linear network I0 on top of the SVD layer S0, and adapt corresponding layer L-I0. The location of insertion is chosen for best tradeoff between accuracy and SD parameters. Adapting L-I0 obviously provides a big benefit in terms of overall number of parameters, as also noted in Table 1. The number of associated parameters is 208 x 208 = 43k, thus a fraction of that required for layer L-D0 and L-O.

Overall the approach of intermediate-layer adaptation along with inserting and adapting a layer provided benefits in terms of both accuracy and SD parameters. This also provides a framework for building future adaptation work. Next, we focus on building an adaptation work in the intermediate-layer framework where we use non-negativity constraints to yield sparser set of SD parameters.

**Table 1**. *Number of required parameters across adaptation techniques. For non-negative techniques we report effective no. of non-zero parameters.*

| Adaptation technique | No. of Parameters (in 1000's) |
|---|---|
| Top layer L-O | 2105 |
| Input layer L-D0 | 1486 |
| Individual layers in (L-S0, L-S1 etc.) | 425 |
| Individual inserted layers in (L-I0 etc.) | 43 |
| Individual inserted layers in (L-I0 etc.) with non-negativity and threshold of 0.03 | 2.2 |

**Table 2**. *Non-negative model adaptation with different threshold constraints, see Sec. 4.2. Baseline (unadapted) WER = 14.15%. Best adapted model without non-negative constraint has WERR = 11.3%.*

| Threshold | Non-zero weights [%] | WERR [%] |
|---|---|---|
| 0 | 72.1 | 11.17 |
| 0.0001 | 60.3 | 11.17 |
| 0.0002 | 46.5 | 10.84 |
| **0.0005** | **13.8** | **10.46** |
| 0.001 | 3.0 | 9.16 |
| 0.005 | 0.5 | 4.55 |

## 4. NON-NEGATIVITE DNN ADAPTATION

In this section we build on the speaker adaptation framework discussed in Sec. 3 and demonstrate a non-negative approach to provide additional strong reduction in the number of SD parameters while still retaining most of the accuracy gain.

Non-negativity constraints have been found to be useful for a number of applications [14]. A non-negative factorization was developed for ASR dereverberation in spectro-temporal domain in [15, 23]. A key benefit of non-negative constraint is that it leads to a sparse set of model parameters, where many of the non-essential parameters are optimized to be identically 0. In Sec. 2.1, we referred to the size of SD parameters as a key challenge in successful deployment of speaker adaptation to large scale tasks. In this context, non-negativity of the adaptation parameters provides an ideal set of constraints towards a small foot-print model. We further describe our non-negative work with respect to Fig. 2. Following the insert and adapt framework in Sec. 3.1, we insert a linear layer L-I0 on top of the SVD layer L-S0. We refer to the weight matrix associated with layer L-I0 as $W_{I0}$, thus $X_{I0} = W_{I0}X_{S0}$, and where $X_{S0}$ and $X_{I0}$ respectively indicate activations for layers L-S0 and L-I0. We initialize, $W_{I0} = I$ *i.e.*, a diagonal matrix with all diagonal entries being 1. This step ensures that the DNN output L-O activations are identical with or without the inserted layer L-I0. At this initialization step the matrix $W_{I0}$ is a non-negative matrix. We follow the standard negative cross entropy (CE) criterion [24] for speaker adaptation:

$$D = \frac{1}{N}\sum_{t=1}^{N}\sum_{s_t=1}^{S}\tilde{p}(s_t|x_t)\log p(s_t|x_t) \quad (1)$$

We update the weight elements $W_{I0}$ in every mini-batch where we also apply the non-negativity constraint in:

$$W_{I0}[i,j] = 0 \quad \forall\, W_{I0}[i,j] < 0 \quad (2)$$

Above approach follows the standard back propagation update with CE criterion except that we additionally force non-negative elements to 0 in each mini-batch of the DNN update. Note that $W_{I0}$ is initialized to be non-negative and it remains non-negative throughout the optimization steps. We note the benefits with this approach in Table 2. We refer to Sec. 4.1 for details on our speaker adaptation experimental setup. In Table 2, we see that non-negativity constraint effectively requires only 72.1% of the parameters in the matrix $W_I$, as the rest are forced to be identically 0. The word-error-rate-relative (WERR) over the baseline (unadapted) is almost identical to that from without non-negativity constraints. A WERR difference of less 1% is expected to be statistically insignificant. Thus this work provides about 27.9% relative reduction in the parameters without any loss in WER over the current best adaptation. In subsequent sections we demonstrate additional scope to reduce the effective number of SD parameters.

### 4.1. Experimental Setup

Our adaptation task consists of Microsoft US English voice-search (VS) data across 50 speakers. Adaptation data includes 50 untranscribed utterances (4-5 mins.) per speaker, where we use the SI model to decode and align data against decoded hypotheses. The adapted models were tested on a distinct test set of 50 utterances per speaker. We used our US English server language model with over 400K words for decoding. Our baseline speaker-independent (SI) DNN AM was trained from a 1000 hours of VS and short-message-dictation (SMD) data with 66-dim dynamic log-MelFilterbank features and a context window of 11 frames, forming an input vector of 726-dim. DNN had 5 hidden layers with 2048 nodes each, 5 SVD layers with about 208 nodes each [5], along with 6000 output units. The hidden layers apply sigmoid nonlinearity; output layer applies softmax. We regularize adaptation with Kullback-Leibler-divergence (KLD) with a regularization coefficient of 0.5 [12]. Our baseline SI model provided a word error rate (WER) of 14.15%. Our previous best adaptation approach without non-negativity constraints provided a WER of 12.55%, thus a WER-relative (WERR) of 11.3%. We use 2-Bytes to represent the SD parameters, thus the current adaptation approach where we adapt an intermediate-layer with 208x208 (43 K) parameters requires 86 KB per speaker. We experimented with adapting different intermediate-layers and found it effective to adapt the inserted layer on top of the $4^{th}$-SVD layer. Our objective in this work is obtain an order an magnitude reduction in the size of SD parameters.

### 4.2. Non-negative Adaptation with a Positive Threshold

Our initial work with non-negativity constraints already provided substantial reduction in the no. of SD parameters. Here we seek an even stronger improvement in the non-negativity framework we discussed. We analyze the update equation (2); we note that choosing a small-positive threshold can provide an additional leverage to yield even sparser models.

$$W_{I0}[i,j] = 0 \quad \forall\, W_{I0}[i,j] < t, where,\ t > 0 \quad (3)$$

Compared to (2), the only modification in (3) is a small-positive $t$. Note that above constraint is applied in each mini-batch of DNN update. This allows DNN to account for some of the approximation loss in above by appropriately updating parameters in subsequent mini-batches. We present the non-zero percent of parameters in $W_{I0}$ for different $t$ in Table 2. There we also include WERR results. As

**Table 3**. *Non-negative model truncation with different non-negative thresholds, see Sec. 4.3. Best adapted model without non-negative constraint has WERR = 11.3%.*

| Threshold | Non-zero weights [%] | WERR |
|---|---|---|
| 0 | 72.1 | 11.17 |
| 0.01 | 27.8 | 11.06 |
| 0.02 | 12.3 | 11.23 |
| **0.03** | **5.1** | **10.19** |
| 0.04 | 2.2 | 8.51 |

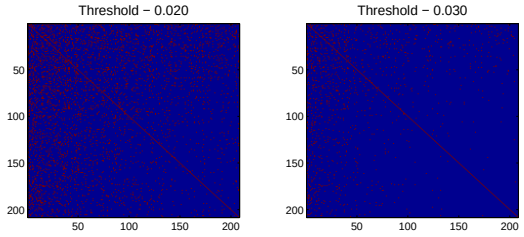


**Fig. 3**. *Non-negative masks for respective thresholds following work in Sec. 4.3. Please zoom-in and print in color.*

expected we see that the non-zero elements decreases substantially with larger $t$; correspondingly WERR too gradually reduces. One of our key performance benchmark for adaptation is a 10+% WERR. Thus we we see that a threshold of $0.0005$ meets our performance objective and reduces the effective no. of parameters to only about 13.8% of that without non-negativity constraints.

### 4.3. Setting Small-Positive Weights in the Non-Negative Model to Zero

In this section, we seek an additional degree of freedom in our speaker adaptation design where we may trade off between WERR and no. of parameters without having to explicitly retrain the SD model. This freedom is essential in an evolving speaker adaptation implementation. Due to resource and cost considerations, an initial implementation may have a far stronger constraint on the size of SD parameters, that we may gradually relax with additional resources at a later stage. In this context, we do not expect to completely retrain SD models for different sizes of SD parameters. It is desirable to build a single set of SD parameters and a new recipe to easily yield varying size of SD parameters, of course with corresponding WER tradeoffs. We demonstrate our non-negative approach to be very suitable for this task. Following the non-negative updates in (2), we first build a sparse set of SD parameters, we denote this in $W_{I0}^0$, here that superscript 0 indicates threshold 0 in (2). Then we apply following modification for a sparser set of parameters:

$$W_{I0}^t[i,j] = 0 \quad \forall\, W_{I0}^0[i,j] < t, where,\ t > 0 \tag{4}$$

We report results with this approach in Table 3. Similar to our earlier observations in Table 2, we obtain trade offs between effective no. of non-zero parameters and WERR. In particular we obtain a very strong operating point with $t = 0.03$, where we require only 5% of non-zero parameters and obtain 10+% WERR. In Fig. 3, we plot the masks for respective thresholds following (4), understandably the non-negative values are squeezed towards upper-left corner due to underlying SVD structure. It is also understandable that the thresholds in the Tables 2 and 3 have different range. We also conducted a similar experiment for the previous best adaptation technique *i.e.* without non-negativity constraints. There, we set weight elements with absolute values less than a threshold to zero. Though, this could provide some reduction in the effective no. of non-zero parameters, the eventual benefit in accuracy and model-size was no match to that obtained from the combination of non-negativity and thresholding in (4). We can also plan to combine the work in Secs.4.2 and 4.3 but we don't expect additional big gains.

### 4.4. Effectively storing Non-negative SD parameters

In Tables 2 and 3, we noted benefit in terms of the effective non-zero SD parameters. Here we continue to explicitly quantify the storage requirements per speaker in Kilo-Bytes (KB). We propose following 3 approaches to represent the overall set of parameters in $W_{I0}$ that has a dimension of 208x208, (a) Bit-Mask - we prepare a mask of size 208x208-bits, where each bit represents non-negativity of the corresponding matrix element, (b) Table-lookup - we store the pair (index, non-negative value) for all non-negative values; we individually require 2-Bytes for index and the value, (c) Single-blob - we can store both non-negative values and zeros in a single binary blob by leveraging the most-significant-bit (MSB) that conventionally represents sign of a value; we simply write a bit 1 for zero-valued SD parameters and 2-Bytes for non-negative parameters where due to non-negativity MSB will always be 0; this facilitates us to effectively write and read the SD parameters and overall requires 1-bit for zeros and 2-Bytes for non-negative values. These techniques provide a spectrum of choices; the final selection will depend on non-zero fraction in data and potentially other constraints from memory management etc. The non-negative approach in Table 3 with a threshold of $0.03$ and effectively 5.1% non-zero parameters in $W_I$, requires 9.8KB, 8.8KB, and 9.5KB across data storing techniques in Bit-Mask, Table-lookup, and Single-blob, respectively. This is a big win over the current adaptation with 86KB, we have achieved an order of magnitude reduction in the size of SD parameters and can far better scale our service for millions of users. We also note our progress in term of no. of SD parameters in Table 1. Clearly the combination of inserting and adapting layer, non-negativity constraint and thresholding provides a new benchmark in speaker adaptation and requires only 2.2K parameters, whereas prior work required 425-2105K parameters.

## 5. CONCLUSION

In this work we presented an intermediate-layer adaptation framework to seek substantial reduction in the number of SD parameters, while still maximizing accuracy benefit with adaptation. We leveraged the framework in terms of a non-negative constraint, where we provided 2 different approaches to seek substantial reduction in the SD parameters. The approaches presented in this work isn't specific to speaker adaptation. Our techniques can also be applied to other ASR adaptation applications. We have already seen benefits for non-native speakers and expect to see benefits for scenarios across device, environment adaptation etc. The specific layer to be adapted may in general be application specific. Overall, our unsupervised speaker adaptation work provided 90% relative reduction in the size of SD parameters by reducing the SD parameters from an earlier best 86KB to 8.8KB, while still retaining 10+% WERR over unadapted baseline. Based on our results we advocate a greater push towards personalization in future.

## 6. REFERENCES


[1] K. Kumar, C. Liu, K. Yao, and Y. Gong, "Intermediate-layer dnn adaptation for offine and session-based iterative speaker adaptation," in *Interspeech*, 2015.

[2] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, 2012.

[3] G.E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing,*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[4] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *IEEE SLT*, 2012.

[5] J. Xue, J. Li, D. Yu, M. Seltzer, and Y. Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *ICASSP*, 2014.

[6] Pawel Swietojanski and Steve Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 171–176.

[7] S. Dupont and L. Cheboub, "Fast speaker adaptation of artificial neural networks for automatic speech recognition," in *ICASSP*, 2000, pp. 1795–1798.

[8] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*, 2003, pp. 55–59.

[9] P. Karanasou, Y. Wang, M. Gales, and P. Woodland, "Adaptation of deep neural network acoustic models using factorised i-vectors," 2014.

[10] X. Lei, H. Lin, and G. Heigold, "Deep neural networks with auxiliary gaussian mixture models for real-time speech recognition," in *in Proc. ICASSP*, 2013.

[11] Y. Miao, H. Zhang, and F. Metze, "Towards speaker adaptive training of deep neural network acoustic models," 2014.

[12] 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 *ICASSP 2013*, 2013.

[13] C. Liu, Y. Wang, K. Kumar, and Y. Gong, "Investigations on speaker adaptation of LSTM RNN models for speech recognition," in *Intenational Conference on Acoustics, Speech, and Signal Processing*, 2016.

[14] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, 1997.

[15] K. Kumar, *A Spectro-Temporal Framework for Compensation of Reverberation for Speech Recognition*, Ph.D. thesis, Ph.D. Thesis: Dept. of ECE, Carnegie Mellon University, 2011.

[16] Kshitiz Kumar, Ziad Al Bawab, Yong Zhao, Chaojun Liu, Benoit Dumoulin, and Yifan Gong, "Confidence-features and confidence-scores for asr applications in arbitration and dnn speaker adaptation," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[17] Y. Zhao, J. Li, J. Xue, and Y. Gong, "Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data," in *Proc. ICASSP*, 2015, pp. 4310–4314.

[18] Sree Hari Krishnan Parthasarathi, Bjorn Hoffmeister, Spyros Matsoukas, Arindam Mandal, Nikko Strom, and Sri Garimella, "fmllr based feature-space speaker adaptation of dnn acoustic models," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[19] Mei-Yuh Hwang and Xuedong Huang, "Subphonetic modeling for speech recognition," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 174–179.

[20] Douglas Reynolds, Richard C Rose, et al., "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[21] Christopher J Leggetter and Philip C Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[22] Mark JF Gales and Philip C Woodland, "Mean and variance adaptation within the mllr framework," *Computer Speech & Language*, vol. 10, no. 4, pp. 249–264, 1996.

[23] K. Kumar, R. Singh, B. Raj, and R. M. Stern, "Gammatone sub-band magnitude-domain dereverberation for asr," *in Proc. IEEE ICASSP*, 2011.

[24] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, 1988.