Frame-wise Online Unsupervised Adaptation of DNN-HMM Acoustic Model from Perspective of Robust Adaptive Filtering

Ryu Takeda and Kazunori Komatani

The Institute of Scientific and Industrial Research, Osaka University, Osaka, Japan

rtakeda@sanken.osaka-u.ac.jp, komatani@sanken.osaka-u.ac.jp

Abstract

We present a new frame-wise *online* unsupervised adaptation method for an acoustic model based on a deep neural network (DNN). This is in contrast to many existing methods that assume *offline and supervised* processing. We use a likelihood cost function conditioned by past observations, which mathematically integrate the unsupervised adaptation and decoding process for automatic speech recognition (ASR). The issue is that the parameter update of the DNN should be less affected by outliers (model mismatch) and ASR recognition errors. Inspired by the robust adaptive fiter techniques, we propose 1) parameter update control to remove the influence of the outliers and 2) regularization using L2-norm of DNN's posterior probabilities of specific phonemes that are prone to recognition errors. Experiments showed that the phoneme recognition accuracies were improved by a maximum of 6.3 points, with an average error reduction rate of 10%, for various speakers.

Index Terms: speech recognition, acoustic model, unsupervised adaptation, nonlinear adaptive filter

1. Introduction

1.1. Motivation

Our purpose is to develop a personalized spoken dialogue system with *life-long learning ability* that adapts its internal models to each user and unseen situations dynamically [1]. Continuous adaptation of acoustic models (AM) and high-accuracy phoneme recognition are fundamental for such a system to deal with model-mismatched speakers and out-of-vocabulary (OOV) words [2, 3, 4, 5, 6]. *Online* and *unsupervised* adaptation of AM during decoding are an important functions because they match the incremental dialogue strategy [7, 8] and they impose only a light user load for adaptation. The *degree of model mismatch* should be also provided during adaptation because the dialogue strategy can also be adapted to users when the system is aware that a user is hard to recognize. Our target is high-performance AMs based on deep neural networks (DNN).

A few studies have focused on unsupervised and online adaptation for DNN AMs while others assume supervised and batch/offline adaptation. The methods of online transformation of acoustic features include normalization parameter [9] and i-Vector [10]. Although useful information for online adaptation, such as sequential estimation results, is available during the frame-wise decoding process, such information has not been exploited yet. Direct adaptation of DNN parameters, which would perform best, have mainly been discussed in the context of supervised/unsupervised batch processing [11, 12, 13]. The nonlinearity of DNNs and the decoding implementation make online unsupervised adaptation difficult. No attention has been paid to *the degree of model mismatch* in either approach.

Our new approach uses a frame-wise likelihood cost function conditioned by past observations for online unsupervised adaptation. DNN parameters are updated by back-propagation in an unsupervised manner, and the decoding process is embedded in the gradient evaluation. We derive an update rule for the AM of DNN and a hidden Markov model (HMM) [14, 15] for enabling the system to register OOV words. The degree of model mismatch is naturally obtained as *evidence* in the generative model [16]. Since the process of frame-wise online AM adaptation has a blind and nonlinear *adaptive filter* perspective [17], we can utilize its techniques and knowledge.

The problem is to ensure that the parameter update is less affected by model mismatch and recognition errors. If the model mismatch is too large from the *viewpoint of the current model*, the back-propagation feedback is too noisy and degrades the parameters of the DNN. In particular, the learning rate is sensitive to mismatched speakers and their ways of speaking. For example, if an inappropriate learning rate is used for such a speaker, an unsupervised adaptation exhibits recognition errors of specific phonemes, such as short pauses and syllabic nasals. Once the DNN parameters are overfitted to such mis-recognized results, recovery from such parameters is almost impossible.

We propose 1) parameter update control and 2) regularization using L2-norm of DNN's posterior probabilities of specific phonemes. The former eliminates the influence of outliers and model-mismatched data by stopping the parameter update when the value of the cost function is large. This is related to the double-talk problem in the robust adaptive filter field [18, 19, 20]. The latter explicitly avoids overfitting of the recognition to specific phonemes; its concept is similar to the automatic control method [21]. Experiments were conducted using various speakers, including model-mismatched speakers.

Our contributions are as follows.

1. We formulate an online unsupervised adaptation of a DNN-HMM model based on the conditional likelihood
2. We propose a parameter update control and regularization of the DNN's posterior to avoid incorrect adaptation

1.2. Relation To Prior Work

Our approach estimates the state of the HMM (decoding) and the DNN parameters (adaptation) simultaneously and in an unsupervised manner. Here, we review the previous methods from several research view points.

Unsupervised adaptation of DNN AM usually assumes batch/offline processing and has not utilized information from the decoding process. For example, the hidden units model uses a DNN's posterior cost function (not sequential) for adaptation [22]. Online approaches are usually based on feature adaptation, such as a normalization parameter [9] and i-Vectors [10]. A DNN's posterior cost function was used in [10]; does not utilize the decoding information on HMM.

Supervised adaptation of the DNN's AM also assumes offline processing and usually limits the parameters for adaptation, such as, by using a linear input network (LIN) [23] and others [11, 24, 25, 12]. Regularization based on the Kullback-Liebler (KL) divergence [26], and MAP estimation of DNN parameters [27] have been proposed to avoid overfitting. The details are summarized in [22].

The main difference between our work and adaptive filtering is the representation of the state space. Speech recognition deals with discrete states, while the adaptive filter estimates continuous states. For example, a dual Kalman filter simultaneously estimates continuous states and the parameters of the observation process [28, 29, 30]. Note that the DNN part of the DNN-HMM represents the *inverse* observation process of the state-space model.

2. Preliminary

2.1. DNN-HMM Model

A DNN-HMM is a kind of state-space model in which the likelihood is represented by the DNN. Given observations $\mathbf{x}_{1:T} = \{\mathbf{x}_1, ..., \mathbf{x}_T\}$ with length T, their likelihood is formulated as

$$p(\mathbf{x}_{1:T}) = \sum_{s_{1:T}} \prod_{t=1}^{T} \frac{p(s_t|\mathbf{x}_t;\boldsymbol{\Theta})p(\mathbf{x}_t)}{p(s_t)} p(s_t|s_{t-1}) \tag{1}$$

where $s_{1:T} = \{s_1, ..., s_T\}$ are discrete hidden states corresponding to $\mathbf{x}_{1:T}$, and $p(s_t|s_{t-1})$ is the transition probability. $p(s_1|s_0)$ is defined as $p(s_1)$ for notation. The likelihood $p(\mathbf{x}_t|s_t)$ is converted into $p(s_t|\mathbf{x}_t;\boldsymbol{\Theta})p(\mathbf{x}_t)/p(s_t)$ by using Bayes' theorem, and the point-wise posterior probability $p(s_t|\mathbf{x}_t;\boldsymbol{\Theta})$ is modeled by a DNN with a parameter set $\boldsymbol{\Theta}$.

2.2. Decoding and Naive Adaptation Cost

Automatic speech recognition is the problem of searching for a sentence W that maximizes the following posterior probability:

$$p(W|\mathbf{x}_{1:T}) \propto p(W) \sum_{s_{1:T}} p(\mathbf{x}_{1:T}|s_{1:T}) p(s_{1:T}), \tag{2}$$

where $p(\mathbf{x}_{1:T}|s_{1:T})p(s_{1:T})$ is an acoustic score based on Eq. (1), and $p(W)$ is a language score. Here, the state space of $s_{1:T}$ is characterized by the HMM topology and language model. The composed state space is usually represented by a weighted finite state transducer (WFST) [31].

As for the acoustic model, the following posterior probability $p(s_t|\mathbf{x}_{1:t})$ at frame t is evaluated recursively during decoding:

$$p(s_t|\mathbf{x}_{1:t}) = \frac{p(\mathbf{x}_t|s_t)p(s_t|\mathbf{x}_{1:t-1})}{\sum_s p(\mathbf{x}_t|s)p(s|\mathbf{x}_{1:t-1})} = \frac{p(s_t|\mathbf{x}_t)}{p(s_t)} \frac{p(s_t|\mathbf{x}_{1:t-1})}{Z_t} \tag{3}$$

$$p(s_t|\mathbf{x}_{1:t-1}) = \sum_{s_{t-1}} p(s_t|s_{t-1}) p(s_{t-1}|\mathbf{x}_{1:t-1}), \tag{4}$$

where $Z_t = \sum_{s_t} p(s_t|\mathbf{x}_t)p(s_t|\mathbf{x}_{1:t-1})/p(s_t)$ is a normalization factor (evidence), and $p(s_t|\mathbf{x}_{1:t-1})$ is a prior probability obtained from the $(t-1)$-th posterior probability $p(s_{t-1}|\mathbf{x}_{1:t-1})$. The evidence Z_t represents how well the model fits the data [16], and it can be used as a measure of model mismatch. Our insight is to utilize this evidence while *the traditional decoding formulation usually skips this normalization because it does not affect the final recognition results (only relative scores are required).*

Naive unsupervised adaptation uses the cross-entropy as its cost function by regarding a recognition result as a correct label. For example, the posterior of the DNN, $p(\hat{s}_t|\mathbf{x}_t)$, is used in [10, 22]. The most likely state $\hat{s}_t$ can sometimes be expressed as by $\mathrm{argmax}_{s_t} p(s_t|\mathbf{x}_t)$, which is independent of the decoding process.

3. Proposed Method

Our approach consists of a posterior probability estimation using Eqs. (3) and (4) and an update of the DNN parameters, as shown in Fig. 1. Because the language model cannot be used directly during the *frame-wise* decoding because of the time-scale difference, our formulation focuses on the acoustic model. This also avoids the influence of the language-model mismatch.

3.1. Gradient based on Conditional Likelihood

The negative log-likelihood $-\log p(\mathbf{x}_t|\mathbf{x}_{1:t-1})$ is our cost function for the DNN parameters. By considering a Markov process for the state-space model [16], i.e., $p(\mathbf{x}_t|\mathbf{x}_{1:t-1}) = \sum_{s_t} p(\mathbf{x}_t|s_t)p(s_t|\mathbf{x}_{1:t-1})$, the instantaneous cost at frame t can be represented as

$$L_t(\boldsymbol{\Theta}) = -\log \sum_{s_t} \frac{p(s_t|\mathbf{x}_t;\boldsymbol{\Theta})p(\mathbf{x}_t)}{p(s_t)} p(s_t|\mathbf{x}_{1:t-1}), \tag{5}$$

where $p(s_t|\mathbf{x}_{1:t-1})$ is the prior probability also used in the decoding by Eq. (4). Hereafter, we denote the DNN's posterior $p(s_t|\mathbf{x}_t;\boldsymbol{\Theta})$ as P_{s_t} and the prior $p(s_t|\mathbf{x}_{1:t-1})$ as $\alpha(s_t)$.

The gradient of the cost function is required to update parameters by back-propagation. Given that the prior $\alpha(s_t)$ is a fixed constant, i.e., a past result, the gradient becomes

$$\frac{\partial L_t(\boldsymbol{\Theta})}{\partial P_{s_t}} = -\frac{1}{\sum_s \frac{P_s}{p(s)}\alpha(s)} \frac{\alpha(s_t)}{p(s_t)} = -\frac{1}{Z_t}\frac{\alpha(s_t)}{p(s_t)}. \tag{6}$$

Here, $p(\mathbf{x}_t)$ cancels out. Since Z_t becomes large for the model-matched $\mathbf{x}_t$, the feedback is naturally weighted according to each data. Because the evaluation of all the states at each t in real time is time consuming, all probabilities are *approximated* using the current hypotheses during the search, as in a particle filter [32]. N-step gradients are calculated incrementally without updating the parameters for mini-batch processing.

Note that the same gradient can be derived from the Q-function used in an EM algorithm [16]. Given the posterior probability $q_{s_t} = p(s_t|\mathbf{x}_{1:t})$ with the current parameter set $\bar{\boldsymbol{\Theta}}$, the expectation of the negative log joint likelihood becomes

$$\mathbb{E}_{s_t}[-\log \frac{P_{s_t}p(\mathbf{x}_t)}{p(s_t)}\alpha(s_t)] = \mathbb{E}_{s_t}[-\log P_{s_t}] + \text{const.}, \tag{7}$$

where $\mathbb{E}_{s_t}$ represents the expectation operator over s_t using q_{s_t}.

We use $J_t(\boldsymbol{\Theta}) = \mathbb{E}_{s_t}[-\log P_{s_t}]$ as the cost function because the calculation of $p(\mathbf{x}_t)$ can be skipped, and its gradient also matches Eq. (6). Here, J_t represents a kind of similarity between the sequential and DNN's posterior probability. As this cost function is still nonlinear, some iterations may be required for the gradient-based update. Section 3.3 discusses the implementation of the parameter update and state search.

3.2. Update Control and Regularization

Here, we introduce a parameter update control for back-propagation to avoid overfitting to recognition errors and outliers. Because data with a large J_t are usually *outliers from the viewpoint of the current model*, the gradient of such data is noisy, and it degrades the parameters. We thus reduce the influence of such data by stopping parameter update via

$$\frac{\partial J_t(\boldsymbol{\Theta})}{\partial P_{s_t}} = \begin{cases} -\frac{1}{Z_t}\frac{\alpha(s_t)}{p(s_t)} & (J_t(\boldsymbol{\Theta}) < T_c) \\ 0 & (\text{otherwise}) \end{cases} \tag{8}$$

where T_c is a thresholding parameter.

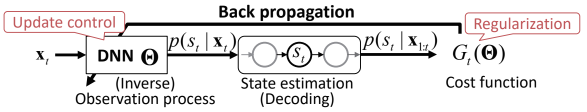

Figure 1: *Overview of our unsupervised adaptation method*

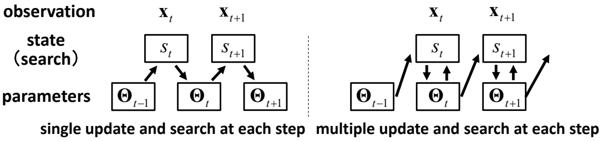
observation
state
(search)
parameters
$\mathbf{x}_t$
$\mathbf{x}_{t+1}$
s_t
s_{t+1}
$\boldsymbol{\Theta}_{t-1}$
$\boldsymbol{\Theta}_t$
$\boldsymbol{\Theta}_{t+1}$
single update and search at each step
multiple update and search at each step

Figure 2: *Two adaptation styles*

We also regularize the DNN's posterior probability with the L2-norm; the following term is added to the cost function: $Q_t = \lambda \sum_{s_t \in M} P_{s_t}^2$, where λ is a weight parameter and M represents a set of states related with specific phonemes, such as a short-pause. This regularization avoids increasing the posterior probability of the states more than is necessary, which is similar to [21]. The whole cost function then becomes $G_t = J_t + Q_t$. Other parameter-based regularizations [26] can be used if they mitigate overfitting, but our regularization directly avoids recognition errors.

3.3. Styles of Decoding and Adaptation

The parameter update based on gradient methods may require iterations because of our actual cost function's non-linearity, and it is not clear how the decoding results are affected. We thus considered two styles of decoding and adaptation, as depicted in Fig. 2.

The first style (on the left side in Fig. 2) updates parameters once per data point, which is like an adaptive filter (pure stochastic gradient) [17]. Its implementation is simple because it involves only three steps: a) posterior probability calculation by Eq. (3) during the search, b) a gradient calculation by back-propagation, and c) parameter update.

The second style (on the right side in Fig. 2) updates parameters several times per data point, which fits parameters more to the local data. Its implementation is a little complicated, because it is necessary to restart the search after each parameter update to calculate the fine posterior probability with the new, updated parameter set. Additionally, if J_t is not increased after the parameter update because of a large learning rate, we must restart the search with the old parameter set and reduce the learning rate by multiplying it by a small value, such as 0.05.

4. Experiment

4.1. Experimental Setups

4.1.1. Data

We conducted experiments with speech data from the Corpus of Spontaneous Japanese (CSJ) [33]. The training set contains 223 hours of academic lecture presentations (by 799 men and 168 women). For the test set, we used the official evaluation sets (eval1, eval2 and eval3) defined in the CSJ, together with our original set (eval4) selected from two-speaker dialogue recordings (6 women and 2 men) in the CSJ. Here, eval4 was a model-mismatch set with respect to the training set in terms of both the acoustic and language models, with speakers sometimes laughing or murmuring in a low voice. The total size of the test set was 6 hours, with 38 speakers in total. 40 principal Japanese phonemes and their BIE (begin, inside, end) expressions were used. The phonemes in the training and the whole test set respectively numbered almost 10 million and half a million.

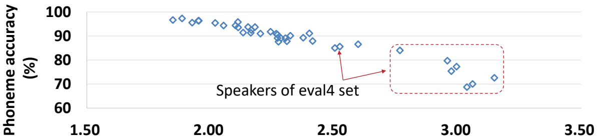

Figure 3: *Joint distribution of log evidence and recognition accuracy for the baseline. Each point denotes a speaker.*

4.1.2. Configurations

Because our aim was to evaluate the adaptation scheme represented by Eq. (6), the structure of the DNN itself needed to match the mathematical model. Our DNN configuration closely followed the one described in [34], because the structure has no global parameters or recurrent structure, which satisfies the model assumptions. As the basic configuration followed [34]; we will only briefly explain it here. It consisted of feature-extraction networks and fully-connected networks. The former network consisted of the mel-filterbank feature extraction, but the parameters were optimized by back-propagation. For example, it used a 10-ms frame shift, fast Fourier transform (512 dim.), absolute, linear projection (filterbank, 64 dim.), absolute, power (instead of log), frame concatenation, and linear projection (bottleneck, 256 dim.) functions. The fully-connected network had seven layers with a sigmoid function, and the number of nodes was 3072. The output layer was a softmax function. Note that *the DNN structure itself is not an important aspect from which to prove the validity of our idea and fundamental formulation.*

The HMM topology and tied-states were obtained by using the Kaldi recipe for the CSJ [35]. The number of tied-states used in the DNN was 9539. Our DNN was trained using a training set with a cross-entropy cost function and online training by Eq. (3) with supervision. AdaGrad [36] was used in both the training and adaptation phases, while keeping the cumulative gradient. The mini-batch size was 32 in both the training and adaptation phases. The prior $p(s_t)$ was assumed to be uniform and set to 1, which did not affect the results in any aspect except scaling.

As constraint-like language models for the phoneme sequences, we used the OpenFST [31] and OpenGrm [37] toolkits to generate (W)FSTs for decoding. An FST without a language score was built by generating all possible Japanese syllable connections (*hiragana*) manually. The WFST for the phoneme N-gram ($N = 8$) was trained by using a phoneme transcription of the training set with 0.5-threshold Seymour pruning. A word language model was not used, because eval4 was an out-of-domain set. The language model weight was set to 1.5, which performed the best.

Speech recognition was performed using our original WFST-based decoder with a 1-best Viterbi search. The continuous parameter update was applied to each speaker independently. The beam size for pruning was 150 (in log-scale), and the maximum number of kept hypotheses was limited to 6,000. To evaluate the pure, sequential adaptation performance, rescoring of the recognition results by using the word graph was not applied. T_c and λ were set to 4.0 and 1.0, respectively. They were selected from a few candidates. We considered /sp/, /q/, and /N/ as *specific* phonemes (non speech frame).

When we use the language model, two decoding processes

Table 1: *Phoneme accuracy (%): only acoustic model*

Set/Spkr	baseline	CE	AF	AF+r	AF+r+u	Itr+r+u
eval1	91.88	92.15	92.19	92.38	92.38	**92.41**
eval2	93.58	94.21	94.35	94.68	94.68	**94.73**
eval3	89.90	90.50	90.62	90.76	90.84	**90.91**
eval4	77.19	77.49	77.82	78.56	**79.30**	79.29
Sum/Ave.	90.13	90.59	90.72	91.04	91.13	**91.17**
SpkrA	87.90	91.91	92.69	94.21	94.20	**94.21**
SpkrB	75.37	74.94	76.10	76.50	76.53	**76.81**
α	–	1.0e-2	6.0e-3	2.0e-2	2.0e-2	6.7e-3

Table 2: *Phoneme accuracy (%): with phoneme 8-gram score*

Set/Spkr	baseline	CE	AF	AF+r+u	Itr+r+u
eval1	93.42	93.41	93.50	**93.64**	93.42
eval2	94.89	95.11	95.38	**95.67**	95.55
eval3	91.47	91.61	91.72	**91.99**	91.91
eval4	77.82	77.14	76.74	79.32	**79.86**
Sum/Ave.	91.48	91.50	91.59	**92.12**	91.93
SpkrA	91.55	94.02	95.65	**96.76**	96.43
SpkrB	73.25	72.12	73.90	76.59	**76.78**
α	–	6.0e-3	6.0e-3	2.0e-2	6.7-e3

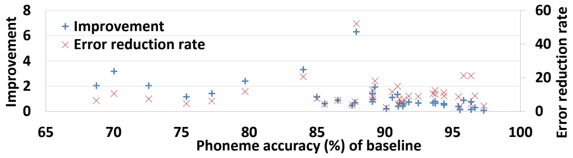

Figure 4: *Improvement of accuracy and ERR for each speaker*

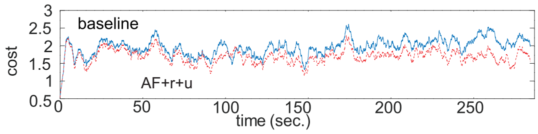

Figure 5: *Smoothed J_t of the baseline and AF+r+u*

run in parallel. One decoder is used for back-propagation of the acoustic model *without any influence from the language score*. The other decoder is used for the hypothesis search using the language score. The DNN posterior P_{s_t} is shared among these two decoders.

4.2. Results and Discussion

First, we found that the evidence functioned as a measurement of the model mismatch, as indicated in Fig. 3, which shows the relationship between the mean negative log evidence $\mathbb{E}_t[-\log(Z_t)]$ and the phoneme recognition accuracy of the baseline for each speaker. Here, the *baseline* means the recognition results without adaptation or language score. The recognition accuracy was widespread, and the evidence was closely correlated with the recognition accuracy. The values of eval4's evidence are obviously different from those of the other sets. We can also say that *the distribution between the speakers and the degrees of model mismatch was a reasonable measure* with which to evaluate the stability of our adaptation method.

Table 1 summarizes the phoneme recognition accuracies of the baseline and our methods for each test set, along with the results for two typical speakers (spkrA in the eval2 set and spkrB in the eval4 set). The learning rates α for the best total accuracy during adaptation are also shown. In the table, *CE* denotes the cross-entropy cost function described in section 2.2, while *AF* and *Itr* respectively denote adaptive filtering and an iterative adaptation style with three iterations. *+r* and *+u* denote methods with regularization and update control, respectively. On average, the accuracies of AF improved by 0.59 and 0.13 from those of the baseline and CE, and AF+r+u additionally outperformed AF by 0.41 on average. In particular, the improvement for eval4 was 1.48, which indicates that update control and regularization are important for model-mismatched conditions such as eval4. Note that CE failed in the case of SpkrB, who spoke about a different topic. Because the gap between AF+r+u and Itr+r+u was small, we conclude that AF-style decoding is an efficient implementation of online adaptation that balances the accuracy and computational cost.

We confirmed that the AM adaption is also effective with constraint-like language models. Table 2 summarizes the phoneme recognition accuracies of the baseline and our methods in terms of the phoneme 8-gram score in parallel decoding. Our adaptation method improved accuracy in all cases, even with an 8-gram score. Note that the accuracy for eval4 with the baseline did not improve much by using this score because of the domain mismatch. While the phoneme accuracy for SpkrB dropped even when the 8-gram was used, the update control and regularization improved accuracy in such mismatched cases.

Figure 4 shows each speaker's performance improvement from the viewpoint of their phoneme accuracy. The horizontal axis indicates the baseline's phoneme accuracy without a language score. The left vertical axis indicates the difference in accuracy between AF+r+u and the baseline, and the right vertical axis indicates the corresponding error reduction rate (ERR). The speaker-mean improvement in the phoneme accuracy was 1.13, with a maximum of 6.3. The speaker-mean ERR rate was 10.29, with a maximum of 52.0. Although the degree of improvement differed among speakers, some ERRs were over 20 even in the region of phoneme accuracies over 95%.

We expect that the performance of our method will improve with more and more data, which is an important aspect for life-long-learning spoken dialogue systems. Figure 5 plots the SpkrB's smoothed J_t of the baseline and AF+r+u at each frame. The horizontal axis indicates the cumulative length of utterances. Note that *the improvements of our method come from the latter parts of the utterace* in the experiments. Since the gap does not seem to converge, continuous adaptation has potential for further improvement.

Although we used the adaptive filtering style (stochastic gradient), a more optimized parameter-update algorithm would be obtained by using a Bayesian framework or nonlinear adaptive filterings such as [27] and Kalman/particle filtering [30, 38, 32]. Because this problem involves nonlinear, unsupervised/blind adaptation, it is a challenging problem. Remaining issues include evaluations of other DNN configurations including recurrent structures and in combination of other adaptation methods.

5. Conclusion

We tackled the problem of unsupervised sequential adaptation of a DNN-HMM acoustic model. We formulated the adaptation by using the conditional likelihood and derived a parameter adaptation rule. We proposed parameter update control and regularization using the L2-norm of the DNN's posterior probability to avoid unstable behavior in adaptation. Our experiments revealed that the method worked and it improved phoneme recognition accuracy by a maximum of 6.3.

Acknowledgement This work was supported by JST, PRESTO Grant Number JPMJPR1857, Japan.

6. References

[1] R. Takeda and K. Komatani, "Attribute prediction of unknown lexical entities based on mixture of bayesian segmentation model," in *Proc. of Life Long Learning for Spoken Language Systems Workshop*, 2019.

[2] K. Ng and V. W. Zue, "Subword-based approaches for spoken document retrieval," *Speech Communication*, vol. 32, no. 3, pp. 157–186, 2000.

[3] G. Neubig, M. Mimura, S. Mori, and T. Kawahara, "Learning a language model from continuous speech," in *Proc. of Interspeech*, 2010, pp. 1053–1056.

[4] L. Qin, M. Sun, and A. I. Rudnicky, "OOV detection and recovery using hybrid models with different fragments," in *Proc. of Interspeech*, 2011, pp. 1913–1916.

[5] T. Shinozaki, S. Watanabe, D. Mochihashi, and G. Neubig, "Semi-supervised learning of a pronunciation dictionary from disjoint phonemic transcripts and text," in *Proc. of Interspeech*, 2017, pp. 2546–2550.

[6] K. Ono, R. Takeda, E. Nichols, M. Nakano, and K. Komatani, "Lexical acquisition through implicit confirmations over multiple dialogues," in *Proc. of Annual SIGdial Meeting on Discourse and Dialogue*, 2017, pp. 50–59.

[7] G. Skantze and A. Hjalmarsson, "Towards incremental speech generation in dialogue systems," in *Proc. of the Annual Meeting of the Special Interest Group on Discourse and Dialogu*, 2010, pp. 1–8.

[8] A. C. Coman, K. Yoshino, Y. Murase, S. Nakamura, and G. Riccardi, "An incremental turn-taking model for task-oriented dialog systems," in *Proc. of Interspeech*, 2019, pp. 4155–4159.

[9] T. Kosaka, H. Yamamoto, M. Yamada, and Y. Komori, "Instantaneous environment adaptation techniques based on fast PMC and MAP-CMS methods," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 1998, pp. 789–792.

[10] H. Arsikere and S. Garimella, "Robust online i-Vectors for unsupervised adaptation of DNN acoustic models: A study in the context of digital voice assistants," in *Proc. of Interspeech*, 2017, pp. 2401–2405.

[11] L. Samarakoon and K. C. Sim, "Factorized hidden layer adaptation for deep neural network based acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2241–2250, 2016.

[12] L. Samarakoon, B. Mak, and K. C. Sim, "Learning factorized transforms for unsupervised adaptation of lstm-rnn acoustic models," in *Proc. of Interspeech*, 2017, pp. 744–748.

[13] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7947–7951.

[14] D. Povey and *et al.*, "The kaldi speech recognition toolkit," in *Proc. of IEEE workshop on automatic speech recognition and understanding*, 2011.

[15] 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," *Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[17] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ 07458: Prentice-Hall, 1991.

[18] K. Ghose and U. Reddy, "A double-talk detector for acoustic echo cancellation applications," *Signal Processing*, vol. 80, pp. 1459–1467, 2000.

[19] T. Gansler, S.L. Gray, M. M. Sondhi, and J. Benesty, "Double-talk robust fast converging algorithm for network echo cancellation," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 6, pp. 656–663, 2000.

[20] T. Waterschoot, G. Rombouts, P. Verhoeve, and M. Moonen, "Double-talk-robust prediction error identification algorithms for acoustic echo cancellation," *IEEE Transactions on Signal Processing*, vol. 55, pp. 846 – 858, 2007.

[21] B. Peterson and K. Narendra, "Bounded error adaptive control," *IEEE Transactions on Automatic Control*, vol. 27, no. 6, pp. 1161–1168, 1982.

[22] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *Proc. of IEEE Spoken Language Technology Workshop*, 2014, pp. 171–176.

[23] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *Proc. of Eurospeech*, 1995, pp. 2183–2186.

[24] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," *Speech Communication*, vol. 49, no. 10, pp. 827–835, 2007.

[25] S. Xue, O. Abdel-Hamid, H. Jian, L. Dai, and Q. Lui, "Fast adaptation of deep neural network based on discriminat codes for speech recognition," *IEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1713–1725, 2014.

[26] 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 *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7893–7897.

[27] Z. Huang, S. M. Siniscalchito, I-F. Chen, W. Jiadong, and C.-H. Lee, "Maximum a posteriori adaptation of network parameters in deep models," in *Proc. of Interspeech*, 2015, pp. 1076–1080.

[28] S. Singhal and L. Wu, "Training multilayer perceptrons with the extended kalman algorithm," in *Proc. of the 1st International Conference on Neural Information Processing Systems*, 1988, pp. 133–140.

[29] E. A. Wan and A. T. Nelson, "Dual kalman filtering methods for nonlinear prediction, smoothing, and estimation," in *Proc. of Neural Information Processing Systems 9*, 1997.

[30] S. Haykin, *Kalman Filtering And Neural Networks*. Wiley Online Library, 2001.

[31] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: A general and efficient weighted finite-state transducer library," in *Proc. of International Conference on Implementation and Application of Automata*, ser. Lecture Notes in Computer Science, vol. 4783. Springer, 2007, pp. 11–23.

[32] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," in *Proc. of International Conference on Neural Information Processing Systems*, 2000, pp. 563–569.

[33] K. Maekawa, "Corpus of spontaneous Japanese: Its design and evaluation," in *Proc. of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[34] R. Takeda, K. Nakadai, and K. Komatani, "Multi-timescale feature-extraction architecture of deep neural networks for acoustic model training from raw speech signal," in *Proc .of IEEE/RSJ International Conference on Intelligent Robots and System*, 2018, pp. 2508–2510.

[35] "Kaldi csj egs," https://github.com/kaldi-asr/kaldi/blob/master/egs/csj/s5/.

[36] J. Duchi, Elad. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[37] B. Roark, R. Sproat, C. Allauzen, M. Riley, J. Sorensen, and T. Tai, "The opengrm open-source finite-state grammar software libraries," in *Proc. of ACL 2012 System Demonstrations*, 2012, pp. 61–66.

[38] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proc. of IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.