Restructuring Output Layers of Deep Neural Networks using Minimum Risk Parameter Clustering

Yotaro Kubo [1]*, Jun Suzuki, Takaaki Hori, Atsushi Nakamura* [2]

NTT Communication Science Laboratories

yotaro@ieee.org, {suzuki.jun, hori.t}@lab.ntt.co.jp, atsushi@nsc.nagoya-cu.ac.jp

Abstract

This paper attempts to optimize a topology of hidden Markov models (HMMs) for automatic speech recognition. Current state-of-the-art acoustic models for ASR involve HMMs with deep neural network (DNN)-based emission density functions. Even though DNN parameters are typically trained by optimizing a discriminative criterion, topology optimization of HMMs is usually performed by optimizing a generative criterion. Several approaches have been studied to achieve a discriminative state clustering, these approaches typically assume underlying Gaussian distributions of the acoustic features, and do not compatible with DNN-based emission density functions. In this paper, we attempt to derive a discriminative restructuring method of an HMM topology by introducing discriminative optimization with discrete constraints on the parameters, which force the parameters to be tied with the parameters of the other states. By applying this constrained optimization to the clustering of parameters of DNN-based acoustic models, we derived a discriminative HMM restructuring method that maintains discriminative performance of the original HMMs with the large number of states.

Index Terms: Automatic speech recognition, context clustering, discrete constraint optimization

1. Introduction

Topology optimization of hidden Markov models (HMMs) is one of the central problems in automatic speech recognition (ASR) since modifying the number of states enables control of several trade-offs including a trade-off between computational efficiency and recognition accuracy. Since optimization of HMM topologies is important for obtaining a reasonable representation of context-dependent phones, this optimization has been studied as a method for context clustering. Context clustering is typically realized by splitting a group of context-dependent phones to several clusters by using phonetic questions [1, 2, 3]. On the other hand, minimizing the number of states while keeping the accuracy is also considered as an important problem since reducing the number of states in HMMs is important to improve generalization ability and to reduce computational complexity of decoding [4, 5, 6]. The trade-offs with regard to computational complexity are especially important for deploying ASR technologies to mobile devices.

However, the current mehtods used for topology optimization do not directly reflect the state-of-the-art techniques for acoustic modeling. Specifically, conventional state clustering and restructuring methods optimize likelihoods [1], variational Bayesian lower bounds [2], KL divergence [5], or linear discriminant function [3] computed by assuming Gaussian distributions even though a current state-of-the-art acoustic model is based on deep neural networks (DNNs) trained by a discriminative criterion. Some methods are capable of handling DNNs directly [6], or via log-linear models [4]. However, these methods are not capable of optimizing discriminative criteria.

This paper introduces a flat direct optimization method for discriminative HMM state clustering. We first introduce a general framework to perform clustering and empirical risk minimization jointly; and then the method is applied for restructuring output layers of DNNs used in acoustic models. Unlike other methods, the proposed mathematical framework can be applied to arbitrary kinds of emission density distributions. In this paper, as a first attempt, we apply the proposed method to a restructuring of log-linear models that involve independent parameter vectors for each arc in decoding network, and feature extraction function based on DNNs [7]. To verify the efficiency of the proposed method, we performed the state clustering experiments on the TIMIT continuous phoneme recognition task [8], and evaluated the phoneme error rates.

2. Minimum Risk Parameter Clustering

In this section, we introduce a mathematical framework used for clustering. Hereafter, we consider that a training dataset is given as $\mathcal{D} \stackrel{\text{def}}{=} \{(\mathbf{X}^{(1)}, \boldsymbol{q}^{(1)}), (\mathbf{X}^{(2)}, \boldsymbol{q}^{(2)}), \cdots\}$ where $\mathbf{X}^{(n)}$ is the n-th sequence of speech features in the training dataset, and $\boldsymbol{q}^{(n)}$ is the reference state sequence corresponding to $\mathbf{X}^{(n)}$. We assume that the model parameter Θ is given as a set of vectors representing the parameters of each HMM state as $\Theta = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots \boldsymbol{\theta}_s \cdots \boldsymbol{\theta}_S\}$ where s denotes an HMM state, and S is the number of HMM states in the original (non-clustered) HMM. The training of the model parameters Θ is generally performed by minimizing a empirical risk function denoted as $R(\Theta; \mathcal{D})$, as follows:

$$\underset{\Theta}{\text{minimize}}\ R(\Theta; \mathcal{D}). \tag{1}$$

Typically, $R(\Theta; \mathcal{D})$ is designed to approximate an amount of classification errors produced by the model parameter Θ.

A joint optimization of clustering and classification can be formulated by introducing constraints that force the HMM state parameters $\boldsymbol{\theta}_s$ to be chosen from a set of the clustered parameters $\bar{\boldsymbol{\theta}}_c$, as follows:

$$\underset{\Theta, \bar{\Theta}}{\text{minimize}}\ R(\Theta; \mathcal{D}) + \Omega(\bar{\Theta}),\ \text{ subject to } \boldsymbol{\theta}_s \in \bar{\Theta} \tag{2}$$

where $\bar{\Theta}$ is a set of clustered parameters $\bar{\Theta} = \{\bar{\boldsymbol{\theta}}_1, \bar{\boldsymbol{\theta}}_2, \cdots \bar{\boldsymbol{\theta}}_C\}$, Ω represents a regularization function for the clustered parameters, and C is the number of the clustered HMM states. Even though this optimization problem is straightforwardly representing empirical risk minimization with the clustered parameters, solving this optimization directly is difficult in general since it involves a mix of combinatorial and continuous optimization.

To solve this optimization, we adapt the method proposed in [9], which is developed based on dual decomposition and the

[1] The author is currently with amazon.

[2] The author is currently with Nagoya City University.

alternating direction method of multipliers (ADMM) [10, 11]. The ADMM [10, 11] is a general framework to solve an intractable optimization by decomposing the optimization problem of the parameter set into several optimization problems of several parameter sets where all the parameter sets are constrained by equality constraints. [9] showed that the ADMM can also be applied to the optimization problem with discrete parameter values. We follow this discrete optimization method, and extend it to the optimization problem with vector-quantized parameter values.

By introducing an auxiliary parameter variable $\Theta' \stackrel{\text{def}}{=} \{\boldsymbol{\theta}'_1, \boldsymbol{\theta}'_2, \cdots \boldsymbol{\theta}'_S\}$ constrained as $\boldsymbol{\theta}'_s = \boldsymbol{\theta}_s$ $(\forall s)$, the original optimization problem (Eq. (2)) can equivalently be transformed to the following problem:

$$\begin{aligned} &\underset{\Theta,\bar{\Theta}}{\text{minimize}}\ R(\Theta;\mathcal{D}) + \Omega(\bar{\Theta}), \\ &\text{subject to}\ \boldsymbol{\theta}_s = \boldsymbol{\theta}'_s,\ \ \boldsymbol{\theta}'_s \in \bar{\Theta}. \end{aligned} \tag{3}$$

This formulation is an extension to the method proposed in [9], which considers multivariate clustering instead of scalar quantization with fixed codes.

Aiming for applying the ADMM, an augmented Lagrange function of the above optimization is derived by introducing Lagrange multipliers $\Lambda \stackrel{\text{def}}{=} \{\boldsymbol{\lambda}_s | \forall s\}$, as follows:

$$\begin{aligned} L(\Theta, \Theta', \bar{\Theta}, \Lambda) =& R(\Theta) + \Omega(\bar{\Theta}) \\ &+ \sum_s (\boldsymbol{\lambda}_s)^\mathsf{T} \boldsymbol{g}(\boldsymbol{\theta}_s, \boldsymbol{\theta}'_s) + \frac{\mu}{2} \sum_s ||\boldsymbol{g}_s(\boldsymbol{\theta}_s, \boldsymbol{\theta}'_s)||^2, \end{aligned} \tag{4}$$

where $\mu > 0$ is a penalty parameter, $\boldsymbol{g}$ is a vector function representing the parameter constraints designed so that $\boldsymbol{g}(\boldsymbol{\theta}_s, \boldsymbol{\theta}'_s) = \mathbf{0}$ iff $\boldsymbol{\theta}_s = \boldsymbol{\theta}'_s$. In the proposed method, $\boldsymbol{g}$ is defined as follows:

$$\boldsymbol{g}_s(\boldsymbol{\theta}_s, \boldsymbol{\theta}'_s) = \gamma_s(\boldsymbol{\theta}_s - \boldsymbol{\theta}'_s), \tag{5}$$

where γ_s is an state-wise weight for constraints.

Following the standard procedure of the ADMM, we attempt to find a local optimum of Θ, Θ', and $\bar{\Theta}$ by finding a saddle point of the augmented Lagrange function. A saddle point of the augmented Lagrange function can be obtained by solving the following nested optimization:

$$\begin{aligned} &\underset{\Theta,\Theta',\bar{\Theta}}{\text{minimize}}\ \underset{\Lambda}{\text{maximize}}\ L(\Theta, \Theta', \bar{\Theta}, \Lambda), \\ &\text{subject to}\ \boldsymbol{\theta}'_s \in \bar{\Theta}. \end{aligned} \tag{6}$$

We should note that the constraints $\boldsymbol{\theta}'_s \in \bar{\Theta}$ are still needed to be satisfied in the above optimization. However, this optimization is more tractable since Θ is not constrained, and Θ' is appeared only in quadratic or linear terms of the augmented Lagrange function.

The optimization defined in Eq. (6) can be solved by alternating the target variable of optimization. With fixed Θ, Θ', and $\bar{\Theta}$, an update rule of the Lagrange multipliers Λ can be derived based on the gradient ascent method, as follows:

$$\boldsymbol{\lambda}_s \leftarrow \boldsymbol{\lambda}_s + \xi \boldsymbol{g}_s(\boldsymbol{\theta}_s, \boldsymbol{\theta}'_s), \tag{7}$$

where ξ is a learning rate for this update.

The update of the variable Θ can be derived as follows:

$$\begin{aligned} \Theta \leftarrow& \underset{\Theta}{\text{argmin}} R(\Theta;\mathcal{D}) + \sum_s (\boldsymbol{\lambda}_s)^\mathsf{T} \boldsymbol{g}(\boldsymbol{\theta}_s, \boldsymbol{\theta}'_s) \\ &+ \frac{\mu}{2} \sum_s ||\boldsymbol{g}_s(\boldsymbol{\theta}_s, \boldsymbol{\theta}'_s)||^2, \\ =& \underset{\Theta}{\text{argmin}} R(\Theta;\mathcal{D}) + \frac{\mu}{2} ||\boldsymbol{\theta}_s - (\boldsymbol{\theta}'_s - \frac{1}{\mu}\boldsymbol{\lambda}_s)||^2. \end{aligned} \tag{8}$$

Algorithm 1 Minimum risk parameter clustering via the ADMM

Optimize Θ by solving Eq. (1) for obtaining an initial value
$\boldsymbol{\lambda}_s \leftarrow \mathbf{0}$
repeat
  Run k-means clustering to minimize Eq. (9) and determine $\bar{\Theta}$ and Θ'
  $\boldsymbol{\lambda}_s \leftarrow \xi\gamma_s(\boldsymbol{\theta}_s - \boldsymbol{\theta}'_s)$
  Update Θ by solving the optimization in Eq. (8)
until cross validation performance is maximized

Since this optimization is a variant of the original empirical risk minimization (Eq. (1)) modified by adding an additional L2-regularization term, Θ-update can also be derived by applying the conventional optimization method, e.g. the gradient descent method.

Minimization of the Lagrange function with respect to Θ' and $\bar{\Theta}$ can be written as follows:

$$\begin{aligned} &\underset{\Theta',\bar{\Theta}}{\text{minimize}}\ \Omega(\bar{\Theta}) + D(\Theta', \bar{\Theta}), \\ &\text{subject to}\ \boldsymbol{\theta}'_s \in \bar{\Theta}, \end{aligned} \tag{9}$$

where

$$\begin{aligned} D(\Theta', \bar{\Theta}) =& \sum_s (\boldsymbol{\lambda}_s)^\mathsf{T} \boldsymbol{g}(\boldsymbol{\theta}_s, \boldsymbol{\theta}'_s) + \frac{\mu}{2} \sum_s ||\boldsymbol{g}_s(\boldsymbol{\theta}_s, \boldsymbol{\theta}'_s)||^2, \\ =& \frac{\mu}{2} \sum_s \gamma_s ||\boldsymbol{\theta}'_s - (\boldsymbol{\theta}_s + \frac{1}{\mu}\boldsymbol{\lambda}_s)||^2 + \text{const.} \end{aligned} \tag{10}$$

Since D is the weighted sum of L2 distances, this optimization is identical with the weighted k-means optimization when $\Omega(\bar{\Theta}) = 0$. Even though we do not derive a tractable solver in general case, we can solve this optimization, if Ω can be written as $\Omega(\bar{\Theta}) = \sum_c \delta ||\boldsymbol{p} - \bar{\theta}_c||^2$, by adding a virtual data point $\boldsymbol{p}$ with a weight δ, which belongs to all clusters.

Algorithm 1 shows the overview of the algorithm, called minimum risk parameter clustering (MRPC), derived by all update rules explained in this section. In the algorithm, we used a solution of the optimization in Eq. (1) as an initial value of Θ in the MRPC, and cross validation is used to stop the optimization process. All update rules are alternatively applied to obtain the result. We should note that each update does not need to minimize the objective function exactly. Performing only one step in the gradient descent method can be used as an update of Θ in the MRPC.

3. DNN State Restructuring via MRPC

In this section, we introduce a method to apply MRPC to restructuring HMM states in DNN-based acoustic models. Even though the general framework proposed in the previous section can be used with DNNs directly, we introduce a method based on log-linear models in this section to derive a computationally efficient method. As a target of the restructuring, we employed a structured log-linear classifier which involves an HMM state for each arc in the weighted finite-state transducer (WFST) representing a decoding network [7]. We first introduce a method to perform training of DNNs as log-linear models, and then we introduce a method to expand the DNN-based acoustic model to the WFST-based log-linear models.

3.1. Neural Nets as Log-Linear Models

Acoustic models based on DNNs compute the conditional probability of the HMM states as follows:

$$P(q_t = s|\boldsymbol{x}_t) = \frac{1}{Z}\exp\left\{\boldsymbol{\theta}_s^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x}_t)\right\}, \quad (11)$$

where q_t is a random variable that denotes the HMM state at time t, $\boldsymbol{x}_t$ is an observed acoustic feature vector at time t, s is a variable denoting the HMM state, Z is a normalization constant, and $\boldsymbol{\phi}(.)$ is the output of the final hidden layer. This equation implies that the DNN acoustic models can also be interpreted as log-linear acoustic models where nonlinearly transformed feature $\boldsymbol{\phi}(\boldsymbol{x}_t)$ is used as an observation vector.

Consider we have an HMM with S states and attempting to reduce the number of states to $C < S$. Applying MRPC is achieved by solving the optimization of Eq. (3) using Algorithm 1 by setting $R(\Theta;\mathcal{D})$ as a discriminative training objective function. This optimization finds a set of C parameters that can minimize the empirical risk. With this application, each state $s \in \{1, 2, \cdots S\}$ in the original HMM is remapped to a state $c \in \{1, 2, \cdots C\}$ where $\boldsymbol{\theta}'_s = \bar{\boldsymbol{\theta}}_c$ in a smaller log-linear model. By transforming the input symbols denoting the original HMM states s to that of the smaller HMM c, a decoding network can further be minimized to reduce the computational complexity for decoding.

The risk function $R(\Theta)$ we used in this paper is derived from the dMMI criterion proposed for discriminative training of HMMs [12, 13], defined as follows:

$$R(\Theta;\mathcal{D}) = \frac{1}{\sigma_2 - \sigma_1}\left(L(\Theta;\sigma_2,\mathcal{D}) - L(\Theta;\sigma_1,\mathcal{D})\right), \quad (12)$$

where L is a modified posterior of the reference state sequence $\boldsymbol{q}^{(n)}$, defined as follows,

$$L(\Theta;\sigma,\mathcal{D}) = \sum_n \log \frac{p(\mathbf{X}^{(n)}, \boldsymbol{q}^{(n)}|\Theta)}{\sum_{\boldsymbol{q}'} p(\mathbf{X}^{(n)}, \boldsymbol{q}'|\Theta) e^{\sigma E(\boldsymbol{q}', \boldsymbol{q}^{(n)})}}. \quad (13)$$

Here, $\boldsymbol{q}^{(n)}$ and $\boldsymbol{q}'$ denote reference and competitor state sequences, respectively, computed by using the original HMM, σ_1 and σ_2 are tunable parameters, the summation $\sum_{\boldsymbol{q}'}$ is computed by using pre-computed lattices, and $E(\boldsymbol{q}', \boldsymbol{q}^{(n)})$ is an error measure between the reference state sequence $\boldsymbol{q}^{(n)}$ and the competitor state sequence $\boldsymbol{q}'$, defined as follows:

$$E(\boldsymbol{q}', \boldsymbol{q}^{(n)}) = \sum_t I\left[q'_t \neq q_t^{(n)}\right], \quad (14)$$

where $I[p]$ is an indicator function that equals to 1 if the given predicator p is true, and equals to 0 otherwise, and q'_t and $q_t^{(n)}$ are t-th elements of $\boldsymbol{q}'$ and $\boldsymbol{q}^{(n)}$, respectively.

In this paper, the above risk function based on dMMI is used in MRPC. However, the framework proposed in the previous section can be integrated with arbitrary objective functions, even with a generative objective function.

3.2. WFST-based State Expansion

MRPC is basically a method for reducing the number of parameters. For applying MRPC to topology optimization of HMMs, we need to define a larger HMM which can be considered as sufficiently expressive. In the following experiments, we used HMMs that involve independent output units for each arc in the decoding network [7]. This section describes a WFST-based definition of DNN-based acoustic modeling.

WFST-based decoders compute the ASR results by solving the following shortest distance algorithm:

$$\underset{\boldsymbol{a},\boldsymbol{t}}{\text{minimize}} \sum_j W[a_j] + \sum_{\tau=t_j}^{t_{j+1}} \omega(a_j, \tau), \quad (15)$$

where $\boldsymbol{a} = \{a_1, a_2, \cdots a_j \cdots\}$ is a sequence of arcs in a decoding network, $\boldsymbol{t} = \{t_1, t_2, \cdots t_j \cdots\}$ is a time alignment, $W[a_j]$ is a transition cost of arc a_j, and $\omega(a_j, \tau)$ is an acoustic cost defined as follows:

$$\omega(a_j, \tau) = -\boldsymbol{w}_{I[a_j]}^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x}_\tau). \quad (16)$$

Here, $I[a_j]$ is an HMM state corresponding to the arc a_j, $\boldsymbol{w}_{I[a_j]}^{\mathsf{T}}$ is $I[a_j]$-th column vector of the last weight matrix of DNN, and $\boldsymbol{\phi}(\boldsymbol{x}_\tau)$ is the output of the last hidden layer computed from the input vector $\boldsymbol{x}_\tau$.

This equation implies that the parameters of the log-linear models are tied according to the HMM states annotated to each arc. We can interpret that each score function $\omega(a_j, \tau)$ is a score of a log-linear model $\boldsymbol{\theta}_s^{\mathsf{T}}\boldsymbol{\phi}(\boldsymbol{x}_\tau)$ with $s = a_j, \theta_s = \boldsymbol{w}_{I[a_j]}$. The WFST-based state expansion method can be defined by untying the parameters of this log-linear model, and directly optimize θ_s instead of optimizing $\boldsymbol{w}_{I[a_j]}$.

The previous studies [7] reported that the use of $\boldsymbol{w}_{I[a_j]}$ as an initial value was important for a training of this WFST expanded model. The method proposed in this paper also follows this training procedure. Therefore, we first train restricted Boltzmann machines as an initial value of DNNs, then perform a back-propagation algorithm for DNN fine tuning, expand the obtained DNNs to a WFST-based parametrization, and apply MRPC for obtaining restructured representation of the WFST-based log-linear model.

4. Experiments

As an initial attempt, we applied the proposed method to the TIMIT continuous phoneme recognition task. We used 11-frames of Mel-frequency cepstral coefficients (MFCC)-based 39 dimensional features computed for each 10 ms as DNN input vectors. Means and variances of the input variables are normalized by using the estimated means and variances computed from the training dataset.

We used 2048 hidden units for each hidden layer, and we inserted a bottleneck layer with 512 hidden units just before the output layer. Using bottleneck layer is advantageous in terms of computational costs for training, and stability of k-means clustering in MRPC. The activation functions of the hidden and bottleneck layers were set to the sigmoid function. Optimization of the monophone DNNs was performed in a similar way to that described in [14]. The slight difference with [14] was that we used 144 monophone HMM states corresponding to 48 reduced phoneme sets as an output class, we halved the learning rate when the frame error rate of the development dataset was increased, and we did not revert to the previous parameters when the validation error increased. Regardless of these small changes, we confirmed that the DNNs' efficiency was also valid under our experimental conditions.

As a method to obtain the baseline triphone model, we applied a method based on variational Bayesian estimation and construction (VBEC) [2]. The number of HMM states obtained by VBEC was 834. The decoding network is constructed as a composition of the WFSTs representing above 834 states and phoneme bigram. The number of arcs in the decoding network was 3436. Therefore, in the experiments, the proposed method was applied to reduce the number of HMM states from $S = 3436$ to the predefined numbers C.

Table 1: Phoneme error rates of the compared methods

Method	# states	Dev.	Eval
monophone	144	21.8	23.1
VBEC [2]	834	20.6	21.9
WFST [7]	3436	19.9	21.1
WFST $\rightarrow$ MRPC	834	20.2	21.2
WFST $\rightarrow$ MRPC	1600	20.0	21.2

The weight for each constraint (γ_s in Eq. (5)) was designed so that the unseen HMM states were not considered in the k-means algorithm in MRPC, i.e. $\gamma_s = 1$ for all s that were appeared in the training dataset, and $\gamma_s \rightarrow 0$ for all s that were not appeared in the training dataset. The hyper parameter in the risk function (σ_1 and σ_2) and the lattice smoothing factor κ were taken from [7], i.e. $\sigma_1 = 1, \sigma_2 = -1$ and $\kappa = 0.2$. The other hyper parameters and the number of iterations of MRPC were tuned by minimizing the phoneme error rates evaluated by the development set. The obtained hyper parameters were $\xi = 1.0$ and $\mu = 100.0$. As initial values of the parameters Θ, we used the baseline DNN parameters as described in [7], and optimized the parameters by using MRPC. The initial value of centroids $\bar{\Theta}$ was set by the initial parameters obtained by a result of cross-entropy training of DNNs with the VBEC-based state alignment. The update of Θ in the algorithm is performed by running single pass of Rprop [15].

To analyze the relation between the number of clusters and the phoneme error rates, we first evaluated the proposed method by varying the number of clusters. We varied C to be one of $\{600, 800, 834, 1000, 1200, 1400, 1600, 2000, 2400\}$ where 834 was equivalent to the number of cluster obtained by the VBEC approach. Figure 1 shows the phoneme error rates (PERs) obtained by varying the number of clustered states. As the other previous studies suggested, the results also exhibit the relative advantage of increasing the number of HMM states from that determined by the generative criterion. However, we also observed that the phoneme error rates saturated after increasing the number of states to 1600.

Table 1 shows the phoneme error rates of the compared method. We confirmed that the proposed method outperformed the conventional parameter estimation method based on VBEC and the cross-entropy criterion. This might be due to two main advantages of the proposed method. The first advantage is that the state clustering is determined so that the discriminative criterion was maximized, and the second advantage is that the log-linear part of the proposed model is trained by a sequential discriminative method. Since the proposed method with 1600 states achieved a comparable result with the full WFST-based log-linear model with 3436 states, the discriminative state restructuring was confirmed to be efficient.

For understanding how MRPC works, we discuss about the convergence of optimization in the proposed method. Figure 2 shows the phoneme error rate as a function of the number of iterations in MRPC optimization. The phoneme error rate of the first iteration in the figure indicates the performance of the parameter vectors obtained by applying k-means clustering to the original log-linear models. Even though the initial value of Θ was obtained by minimizing the risk function, the first k-means clustering step degraded the accuracy of the models. As a result, a performance gap between Θ (the black line in the Figure) and Θ' (the leftmost blue point) was observed. However, by iterating the MRPC steps, it was observed that the performance gap became smaller iteratively. Thus, it was shown that the proposed method performed iterative update of clustering rule for maximizing the performance.

5. Conclusion

In this paper, we proposed a method to restructure the parameters of log-linear acoustic models, and applied this method for minimizing the number of hidden Markov model (HMM) states of the deep neural network (DNN)-based acoustic models. We confirmed that the proposed method achieved minimization of the performance degradation caused by reducing the number of HMM states, and the performance improvement from the triphone clustering determined by a generative method based on Gaussian mixture models.

In future, we will investigate about application of the proposed clustering method to realize a triphone clustering without Gaussian models. This would be achieved by applying the proposed method to very large triphone models that cover all possible triphones. Furthermore, applying the proposed method to large vocabulary continuous speech recognition would also be promising.

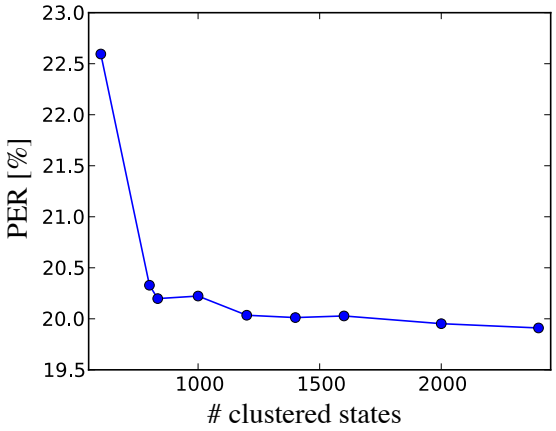

Figure 1: Phoneme error rate of the development set as a function of the number of clustered HMM states

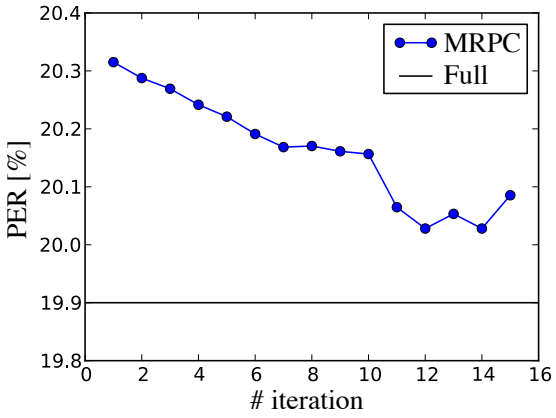

Figure 2: Comparison of the development set performance between fully expanded models and MRPC models

6. References

[1] S. J. Young, J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. the workshop on Human Language Technology*, 1994, pp. 307–312.

[2] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda, "Variational Bayesian estimation and clustering for speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 12, no. 4, pp. 365–381, 2004.

[3] S. Wiesler, G. Heigold, M. Nußbaum-Thom, R. Schlüter, and H. Ney, "A discriminative splitting criterion for phonetic decision trees," in *Proc. INTERSPEECH*, 2010, pp. 54–57.

[4] P. L. Dognin, J. R. Hershey, V. Goel, and P. A. Olsen, "Restructuring exponential family mixture models," in *Proc. INTERSPEECH*, 2010, pp. 62–65.

[5] X. Cui, J. Xue, X. Chen, P. A. Olsen, P. L. Dognin, U. V. Chaudhari, J. R. Hershey, and B. Zhou, "Hidden Markov acoustic modeling with bootstrap and restructuring for low-resourced languages," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 8, pp. 2252–2264, 2012.

[6] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," in *Proc. Interspeech*, 2013.

[7] Y. Kubo, T. Hori, and A. Nakamura, "Integrating deep neural networks into structural classification approach based on weighted finite-state transducers," in *Proc. INTERSPEECH*, 2012.

[8] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 11, pp. 1641–1648, 1989.

[9] J. Suzuki and M. Nagata, "Supervised model learning with feature grouping based on a discrete constraint," in *Proc. ACL*, 2013, pp. 18–23. [Online]. Available: http://www.aclweb.org/anthology/P13-2004

[10] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[12] E. McDermott, S. Watanabe, and A. Nakamura, "Discriminative training based on an integrated view of MPE and MMI in margin and error space," in *Proc. ICASSP*, 2010, pp. 4894–4897.

[13] A. Nakamura, E. McDermott, S. Watanabe, and S. Katagiri, "A unified view for discriminative objective functions based on negative exponential of difference measure between strings," in *Proc. ICASSP*, 2009, pp. 1633–1636.

[14] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[15] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The Rprop algorithm," in *Proc. IEEE ICNN*, 1993, pp. 586–591.