TWO-STAGE SPEAKER ADAPTATION OF HYBRID TIED-POSTERIOR ACOUSTIC MODELS

Jan Stadermann, Gerhard Rigoll

Institute for Human-Machine Communication
Technische Universität München
Arcisstrasse 21, 80290 München, Germany
Phone:+49-89-289-{28319, 28541},
Email: {stadermann, rigoll}@mmk.ei.tum.de

ABSTRACT

For Gaussian distribution-acoustic models there exist many established technologies for speaker adaptation. Contrary to that, there are almost no well-functioning adaptation methods for hybrid systems, consisting of a combination of HMMs and neural networks. In this paper, strategies are explored to adapt hybrid NN/HMM systems based on the tied-posterior paradigm. We investigate the retraining of selected important parts of the neural network and a gradient based adaptation strategy for the HMM's mixture coefficients based on maximizing the scaled likelihood. The paper presents the following innovations: First it introduces one of the first adaptation methods for hybrid systems where the HMM component contributes significantly to the adaptation success. Second, it presents a novel approach to the neural network's adaptation, based on the selection of suitable neurons for adaptation.

Results on the WSJ speaker adaptation test show the capability of our methods to adapt to new speakers especially in case of adapting the neural net and that both methods can be combined to achieve additional improvement of the word error rate in most cases.

1. INTRODUCTION

Today's automatic speech recognition systems are generally trained to be speaker independent. Speaker independence is a key feature to build commercial ASR systems, since no customer wants to accomplish a complete training of acoustic models to get a speaker dependent system. The latter often performs better, so speaker adaptation was found to fill in the gap. Although not as good as speaker-dependent models, an adapted system comes close to speaker-dependent error rates with much less effort.

Well established techniques for speaker adaptation are maximum likelihood linear regression (MLLR) [1] and maximum a-posteriori estimation (MAP) [2].

The MLLR approach is usually applied to Gaussian HMMs and tries to estimate one or several matrices that rotate and shift the distributions' mean values to match the new speaker. Since MLLR maximizes only the likelihood of each distribution separately, it can be extended by a discriminative adaptation procedure called scaled likelihood linear regression (SLLR) [3]. MAP adaptation usually tries to incorporate *a-priori* knowledge about the parameters in the training process resulting in adapted models that consist of some combination of the base model parameters and trained parameters from the adaptation data.

When adapting hybrid NN/HMM systems the methods described above are generally not applicable and new strategies have to be developed. In [4] different techniques for adapting neural networks are compared, like retraining or adding new layers to the network, in [5] additional speaker space units are added to the network and trained maximizing the acoustic likelihood. The basic technology explored in this paper is a hybrid NN/HMM acoustic model with certain advantages compared to Gaussian models: the easy extension of input context, the capability to model arbitrary distributions and the discriminative training procedure. Using the tied-posterior approach first presented in [6] to combine the neural net (NN) and the hidden-Markov models we are able to combine HMM-based adaptation algorithms with NN adaptation strategies. In the following sections we introduce methods for adapting hybrid NN/HMM systems using the tied-posterior approach. First we describe this hybrid system in section 2, then we present one method for adapting the neural net in section 3 and one method for adapting the HMMs in section 4. Results are given in section 5 for supervised adaptation on the WSJ S3-C2 and WSJ S3-P0 speaker adaptation tasks.

2. HYBRID ARCHITECTURE

The basic idea of this hybrid architecture is the combination of a posterior probability estimator with HMMs. The network topology suitable for this task is a fully-connected multi-layer perceptron with one hidden layer [7]. We incorporate additional context frames by extending the input layer having an input feature vector $\vec{x} = (\vec{f}(t-m), \ldots, \vec{f}(t), \ldots, \vec{f}(t+m))$. The system used here contains 3 past frames and 3 future frames ($m = 3$). One frame consists of a standard ASR feature vector with 39 components (12 MFCC plus energy and first order and second order derivatives), resulting in an input layer size of 273. The weights are trained with the back-propagation algorithm optimizing the training set's cross entropy, for the output nodes we use the softmax function as non-linearity, the hidden nodes apply the sigmoid function.

To obtain a similar range of value for all input nodes, a normalization process introduced in [8] takes place:

$$x_i^{(n)} = \frac{x_i - \overline{x}_i}{\sqrt{\sigma^2_{x_i}}} \qquad (1)$$

$\overline{x}_i$ is the global mean value for input node i and $\sigma^2_{x_i}$ is the global variance for this node.

The NN is trained to calculate phoneme probabilities. One output node computes the posterior probability $Pr(j|\vec{x})$ that phoneme j has been observed given the (normalized) NN input vector $\vec{x}$. The connection to the HMMs is given by the tied-posterior [6] approach: All HMMs share the NN outputs using separate mixture weights. The set-up is illustrated in figure 1 The probability density value needed for the output of HMM state S_i is computed according to the tied-posterior approach in [6]

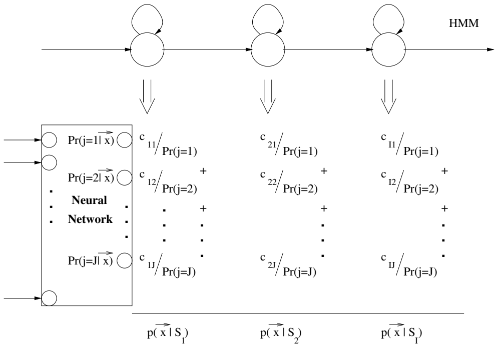

Fig. 1. Tied-posterior architecture

$$p(\vec{x}|S_i) \propto \sum_{j=1}^{J} c_{ij} \cdot \frac{Pr(j|\vec{x})}{Pr(j)} \tag{2}$$

where $Pr(j|\vec{x})$ is the posterior probability and c_{ij} is the mixture coefficient connecting the posterior value $Pr(j|\vec{x})$ with the HMM state S_i.

The *a-priori* probabilities $Pr(j)$ are known in advance and can be computed from the training data.

3. ADAPTING THE NEURAL NET

In [4] several approaches for adapting neural networks are investigated. Best results have been obtained by adding an additional input layer to the NN to adapt the net to new data. Since we have an input layer size of 273 we would have to estimate $273^2 \approx 75000$ parameters which makes the introduction of an additional layer nearly impossible. So we have tried another approach not investigated in [4]: Instead of adding new layers we retrain a sub-set of the hidden units. First we propagate the adaptation data through the original network and select the hidden nodes with maximum activity (see figure 2). Maximum activity is here defined as maximum variance (computed on the adaptation data), since hidden nodes with a high variance transfer a larger amount of information to the output layer. The number of hidden nodes to be selected for adaptation is determined by comparing each node's variance with the maximum value obtained over all hidden nodes over all adaptation data. Pruning takes place if the node's variance is lower than a given percentage of the maximum value. The number of selected nodes after pruning is between 135 and 215 depending on the adaptation data (with fixed pruning threshold for all speakers) corresponding to a number of weights to be adapted between 6345 and 10105. In a second step, these nodes are retrained minimizing the cross entropy between NN outputs and target values. The gradient of weight w_{lj} of one selected hidden node l to be minimized is given as

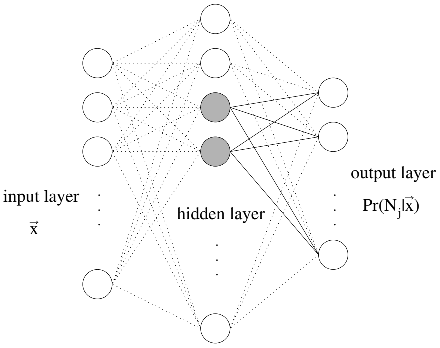

Fig. 2. Unit selection for retraining - filled circles denote selected nodes, thick lines indicate weights to be retrained

$$\frac{\partial E}{\partial w_{lj}} = \sum_{t=1}^{T} \sum_{j=1}^{J} (y_j(t) - \hat{y}_j(t)) z_l(t) \tag{3}$$

where $y_j(t)$ and $\hat{y}_j(t)$ denote the network output and the target value of neuron j and time frame t, respectively, w_{lj} denote the weight from hidden node l to output node j and $z_l(t)$ denotes the activation of hidden node l. The weights are updated using a standard gradient descent procedure with a momentum term. Target values are obtained from the reference transcription, the time alignment has been done with the speaker-independent system. The training is performed with a cross validation set (25% of the adaptation data) where the frame error rate is computed after each iteration. The computation is stopped after a fixed number of iterations and the net with the lowest frame error rate on the cross validation data is taken.

4. ADAPTING THE HMMS

Starting from eq. 2, it is obvious that the HMMs can adjust the likelihood by tuning their mixture coefficients c_{ij}. Therefore, we look for a method to adapt the mixture coefficients and keep the other parts (neural net, prior probabilities, HMM transition probabilities) fixed.
Following [3] we would like to find the HMM parameter λ^* that maximizes the scaled likelihood

$$\lambda^* = \underset{\lambda}{\operatorname{argmax}} \left\{ \frac{p(X|W,\lambda)}{p(X|\lambda)} \right\} , \tag{4}$$

where W denotes a word sequence and X denotes a feature vector sequence. To reduce the computational complexity we apply a frame-wise computation and approximate $p(\vec{x}(t)|\lambda)$ by

$$p(\vec{x}(t)|\lambda) \approx \sum_{S_i \in Q} p(S_i) p(\vec{x}(t)|S_i) \tag{5}$$

In the log-domain we then have to maximize the scaled log-likelihood

$$L = \sum_{t=1}^{T} \log p(\vec{x}(t)|S_v(t)) - \log \left(\sum_{S_i \in Q} p(S_i) p(\vec{x}(t)|S_i) \right) \tag{6}$$

$S_v(t)$ denotes one state of the Viterbi state sequence aligned from the training data and Q denotes the set of all states. Using eq. 2 we could apply a unconstrained gradient ascent procedure, maximizing $\frac{\partial L}{\partial c_{ij}}$.
To ensure that the new coefficients still obey the constraints $c_{ij} \geq 0$ and $\sum_{j=1}^{J} c_{ij} = 1$, we introduce a transformation

$$c_{ij} = \frac{\exp(w_{ij})}{\sum_{k=1}^{K} \exp(w_{ik})} \tag{7}$$

and compute the gradient with respect to w_{ij}:

$$\frac{\partial L}{\partial w_{ij}} = \sum_{t=1}^{T} \delta_{S_v(t)\,i} \frac{c_{ij}\,(Pr(j|\vec{x}(t)) - p(\vec{x}(t)|S_i))}{p(\vec{x}(t)|S_i)} - \frac{Pr(S_i)c_{ij}\,(Pr(j|\vec{x}(t)) - p(\vec{x}(t)|S_i))}{\sum_{S_k \in Q} Pr(S_k)p(\vec{x}(t)|S_k)} \tag{8}$$

($\delta_{S_v(t)\,i}$ denotes the Kronecker delta)

The *a-priori* probability $Pr(S_i)$ is approximated from the adaptation data by computing the relative frequencies of each state.

The gradient ascent is initialized by $w_{ij} = \log c_{ij}$ that results in the speaker independent model parameters as initial values. To keep the number of parameters small, we stick to context-independent monophone models. The phoneme set consists of 45 phonemes plus 2 silence models (`sil` and `sp`) resulting in 47 posterior probabilities and 47 HMMs. The training procedure is performed for a fixed number of iterations. A part of the training set (25%) is kept apart and used for cross validation. After the number of iterations has passed the best weight vector on the cross validation set is taken. The criterion to determine the performance on the cross validation set is the phoneme error rate. In case that less than 2 occurrences of one HMM appear in the adaptation data, the original speaker adaptation model is copied and no adaptation takes places for this model.

5. EXPERIMENTS

The evaluation has been performed on the WSJ S3-C2 test and WSJ S3-P0 test [9] designed for speaker adaptation with native and non-native speakers, respectively. Both tests contain a training set with 10 speakers and 40 sentences each and a test set with between 20 and 43 test utterances per speaker. For all experiments we have used the bigram language model from the WSJ0 database and a dictionary with 5000 words (corresponding to the WSJ hub 2 task). The speaker independent model has been computed using the 7240 sentences from the hub 2 training set. Every 10ms a new frame is shifted in the NN's input layer and the phoneme posterior probability is computed. To observe just the effect of the algorithms presented in sections 3 and 4, the mean and variance values used for feature normalization are not changed with the adaptation data, the mean and variance values from the speaker independent system are used. For the same reason (and to prevent bad *a-priori* estimates from a small amount of data) we do not modify the *a-priori* probabilities $Pr(j)$ that have been computed on the large WSJ hub 2 set.
The adapted system has the same number of parameters as the speaker independent system: The neural net is only retrained and the new mixture coefficients c_{ij} (see section 4) are recalculated after the adaptation is finished. So there is no loss of computation speed in the adapted system. Tables 1 and 2 show results using neural net adaptation and HMM adaptation on their own comparing the word error rate with the unadapted speaker-independent baseline system.

Speaker	**WER SI**	**NN adapt.**	**HMM adapt.**
4OA	5.47%	6.51%	**4.43%**
4OB	7.46%	**5.97%**	7.46%
4OC	10.22%	**8.98%**	10.22%
4OD	10.38%	**9.43%**	12.26%
4OE	16.22%	**14.71%**	**14.71%**
4OF	30.06%	**26.69%**	**29.75%**
4OG	5.41%	6.31%	5.41%
4OH	22.14%	**20.10%**	22.65%
4OI	11.11%	**10.89%**	12.00%
4OJ	34.64%	**28.91%**	35.94%
mean	15.31%	**13.85%**	15.48%

Table 1. WSJ S3-C2 Adaptation results (word error rate) with either NN or HMM adapted compared to the speaker independent baseline error rate (native speakers)

Speaker	**WER SI**	**NN adapt.**	**HMM adapt.**
4ND	36.23%	**30.69%**	**33.58%**
4NE	37.80%	**28.48%**	**29.27%**
4NF	35.96%	**23.45%**	**31.09%**
4NH	34.25%	**23.20%**	**26.38%**
4NI	21.21%	25.04%	**20.94%**
4NJ	22.97%	**12.31%**	**20.05%**
4NK	21.98%	**15.73%**	**16.64%**
4NL	17.40%	**12.38%**	**14.09%**
4NM	36.94%	**33.28%**	**33.12%**
4NN	36.71%	**31.93%**	**36.43%**
mean	30.15%	**23.65%**	**26.16%**

Table 2. WSJ S3-P0 adaptation results (word error rate) with either NN or HMM adapted compared to the speaker independent baseline error rate (non-native speakers)

Tables 3 and 4 give the result of the two-step adaptation. First the neural net is adapted using the method presented in section 3, then the Markov models are adapted using the adapted NN with the technique described in section 4.

Discussing the results one can observe a significant improvement applying the NN adaptation for both sets. The HMM adaptation on its own performs well with non-native speakers but worse with the native-speaker set. The two-stage adaptation (NN adaptation first, then HMM adaptation) improves the system's overall performance on both tests.
It is clear that the HMM adaptation algorithm has difficulties to recover from completely wrong phoneme probabilities delivered by the NN. On the other hand, if the NN delivers several candidates for the correct class (with nearly equal probabilities), the HMM adaptation procedure amplifies the correct one.

6. CONCLUSION

An algorithm for adapting hybrid NN/HMM acoustic models is presented. The models are based on the tied-posterior approach where one NN estimating phoneme posterior probabilities

Speaker	**WER SI**	**NN and HMM adapt.**
4OA	5.47%	7.03% (0.29)
4OB	7.46%	**5.72%** (-0.23)
4OC	10.22%	**8.98%** (-0.12)
4OD	10.38%	**8.02%** (-0.23)
4OE	16.22%	**15.02%** (-0.07)
4OF	30.06%	**26.69%** (-0.11)
4OG	5.41%	5.71% (0.06)
4OH	22.14%	**17.81%** (-0.20)
4OI	11.11%	**11.11%** (0.00)
4OJ	34.64%	**28.91%** (-0.17)
mean	15.31%	**13.50%** (-0.12)

Table 3. WSJ S3-C2 Adaptation results (word error rate) with NN and HMM adapted compared to the speaker independent baseline error rate (numbers in brackets denote the relative deviation)

Speaker	**WER SI**	**NN and HMM adapt.**
4ND	36.23%	**24.15%** (-0.33)
4NE	37.80%	**28.61%** (-0.24)
4NF	35.96%	**23.45%** (-0.35)
4NH	34.25%	**21.69%** (-0.37)
4NI	21.21%	**20.33%** (-0.04)
4NJ	22.97%	**12.31%** (-0.46)
4NK	21.98%	**31.00%** (-0.41)
4NL	17.40%	**11.76%** (-0.32)
4NM	36.94%	**23.89%** (-0.35)
4NN	36.71%	**30.80%** (-0.16)
mean	30.15%	**21.00%** (-0.30)

Table 4. WSJ S3-P0 Adaptation results (word error rate) with NN and HMM adapted compared to the speaker independent baseline error rate (numbers in brackets denote the relative deviation)

is shared by all HMMs. With this architecture it is possible to adapt the NN and the HMM in two separate steps. The NN is adapted by retraining a part of the output layer with a standard gradient descent procedure, the exact number of parameters is selected by the hidden node's variance. The HMM's mixture coefficients are adapted using a gradient based strategy maximizing the scaled likelihood. Experiments performed using the WSJ speaker adaptation test with native and non-native speakers show an improvement in the word error rate. Future research includes the implementation of Eigenvoices and MAP-based adaptation strategies for HMM adaptation and improved selection mechanisms for the retraining of the neural net.

7. REFERENCES

[1] C.J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[2] Jean-Luc Gauvain and Chin-Hui Lee, "Maximum a-posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994.

[3] Frank Wallhoff, Daniel Willett, and Gerhard Rigoll, "Frame Discriminative and Confidence-Driven Adaptation for LVCSR," in *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, June 2000, pp. 1835–1838.

[4] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and A. Robinson, "Speaker-Adaptation for Hybrid HMM-ANN Continuous Speech Recognition System," in *Proc. EUROSPEECH*, Madrid, Spain, Sept. 1995.

[5] Nikko Ström, "Speaker-Adaptation by Modeling the Speaker Variation in a Continuous Speech Recognition System," in *4th Int. Conference on Spoken Language Processing (ICSLP)*, Philadelphia, USA, Oct. 1996.

[6] J. Rottland and G. Rigoll, "Tied posteriors: An approach for effective introduction of context dependency in hybrid NN/HMM LVCSR," in *Proc. ICASSP*, 2000.

[7] Herve Bourlard and Nelson Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, 1994.

[8] Merten Joost and Wolfram Schiffmann, "Speeding up backpropagation algorithms by using cross-entropy combined with pattern normalization," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUKFS)*, vol. 6, no. 2, pp. 117–126, 1998.

[9] F. Kubala, J.R. Bellegarda, J. Cohen, D.S. Pallett, D.B. Paul, M. Phillips, R. Rajasekaran, F. Richardson, M. Riley, R. Rosenfeld, R. Roth, and M. Weintraub, "The Hub and Spoke Paradigm for CSR Evaluation," in *Proc. of the ARPA Spoken Language Technology Workshop*, Plainsboro, New Jersey, Mar. 1994, pp. 9–14, Morgan Kaufmann.