ISCA Archive
http://www.isca-speech.org/archive

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

ROBUST SPEAKER ADAPTATION OF CONTINUOUS DENSITY HMMS USING MULTILAYER PERCEPTRON NETWORK

Mikko Harju[1], *Petri Salmela*[1], *Olli Viikki*[2], *Mikko Lehtokangas*[1] *and Jukka Saarinen*[1]

[1)] Tampere University of Technology,
Signal Processing Laboratory
P.O.Box 553, FIN-33101 Tampere, Finland
Tel: +358-3-365 2111
Fax: +358-3-365 3095
Email: mah@iki.fi

[2)] Nokia Research Center,
Speech and Audio Systems Laboratory
P.O.Box 100, FIN-33721 Tampere, Finland

10.21437/Eurospeech.1999-546

ABSTRACT

The performance of global affine and nonlinear transformations for speaker adaptation in a hidden Markov model (HMM) speech recognition system are compared in this paper. The nonlinear transformation was obtained with a multilayer perceptron network (MLP) which was trained during the adaptation process to transform the mean vectors of the HMMs such that the output probabilities of the HMMs for the adaptation utterances were maximized. The performance of the MLP adaptation method was compared to the maximum likelihood linear regression (MLLR) adaptation procedure. Both of these methods were tested in a connected digit speech recognition system using multi-environment models. The results show that the nonlinear MLP transformation clearly outperforms MLLR in terms of adaptation speed. Moreover, the performance of MLP adaptation with larger amounts of data was comparable to the MLLR performance.

Keywords: Speech recognition, neural networks, speaker adaptation, hidden Markov models

1. INTRODUCTION

The objective of speaker adaptation is to adapt the parameters of a speaker dependent speech recognition system such that the recognition accuracy is maximized for the given speaker. There are several methods already available for re-estimating the model parameters given some limited amount of adaptation data. Classical statistical estimation techniques like maximum a posteriori (MAP) and maximum likelihood (ML) estimation have been used to derive such re-estimation formulas [1, 2, 3]. The ML estimation has also been used to derive a transform-based adaptation method, maximum likelihood linear regression (MLLR) [4]. Moreover, the ML principle has also been used for adapting the means of the HMMs earlier by using a non-linear transformation obtained with the MLP network [5].

In this paper, the non-linear speaker adaptation method of [5] is slightly modified and applied to a noise robust connected digit recognition task. During adaptation, the Viterbi segmentation of the adaptation data is used to train MLP such that the output probability of the HMMs for the adaptation utterrances is maximized. After training, all the mean vectors of the HMMs are transformed with the global MLP. The adaptation method was found to increase both the recognition performance and the noise robustness of the recognizer compared to speaker independent baseline recognition system. The recognition results of the baseline speaker recognition system are compared to the results of both MLLR and MLP adapted recognition systems.

2. MLP ADAPTATION

In this section the derivation of the MLP adaptation method is given and the modifications to the original method are described [5]. The goal of the MLP adaptation procedure is to find a mapping between speaker-independent and speaker-dependent model spaces, i.e. old parameter values are transformed $\hat{\theta} = f_{MLP}(\theta)$ by an MLP network. The estimation of the transformation was approached as maximum likelihood estimation problem where the likelihood $L(O|\hat{\theta})$ of the adaptation utterances O is maximized. The likelihood $L(O|\theta)$ can be expressed as a sum over all possible state sequences $\varsigma \in S$ [4]:

$$L(O;\theta) = \sum_{\varsigma \in S} L(O, \varsigma|\theta) \qquad (1)$$

The speech feature vectors were assumed to be from a first order Markov source, and the HMM state emission probability densities were modelled using mixtures of multivariate Gaussian densities such as the state s was modelled as a sum of M Gaussians pdfs $b_{s,m}$ weighted with $c_{s,m}$ in which $m = \{1, \ldots, M\}$. In this work, only the means $\mu_{s,m}$ of the Gaussian mixture distributions $b_{s,m}$ are transformed, since the state transition probabilities a_{s_1,s_2} and mixture weights affect recognition performance only marginally [4]. Also the variances are left unchanged since robust variance estimation is cumbersome from sparse data [6]. Unlike in [5], no linear transformation in parallel with the MLP was used.

In order to maximize the likelihood in Equation (1), an

auxiliary function is defined [4]:

$$Q(\theta, \hat{\theta}) = \sum_{\varsigma \in S} L(O, \varsigma|\theta) \log L(O, \varsigma; \hat{\theta}) \quad (2)$$

It can be shown that the maximization of the Equation (2) is equivalent of maximizing the Equation (1) [7, 3].

After some manipulation of Equation (2) the contribution of each mixture component of each state to the Equation (2) can be expressed as [4]

$$Q_b(\theta, \hat{b}_{s,m}) = L(O; \theta) \sum_{t=1}^{T} \gamma_{s,m}(t) \log \hat{b}_{s,m}(o_t) \quad (3)$$

where $\hat{b}_{s,m}(o_t)$ is the new estimate of the Gaussian density having mean $\hat{\mu}_{s,m} = f_{MLP}(\mu_{s,m})$ and covariance matrix $\Sigma_{s,m}$. The $\gamma_{s,m}(t)$ is the total occupation probability of state s and mixture m at time t given that the observation sequence ς is generated [4].

The auxiliary function Q can be used as the cost function for the training of the MLP. By differentiating Equation (3) with respect to weight w_{ij} and using the chain rule we get

$$\frac{\partial Q_b(\theta, \hat{b}_{s,m})}{\partial w_{ij}} = \frac{\partial Q_b(\theta, \hat{b}_{s,m})}{\partial f_{MLP}} \frac{\partial f_{MLP}}{\partial w_{ij}} \quad (4)$$

in which $\partial f_{MLP}/\partial w_{ij}$ is given in [8]. This is an appropriate form to be used for standard training algorithms developed for MLPs. When using the Viterbi alignment of the adaptation data, the first partial derivative of Equation (4) simplifies to [4, 9]

$$\frac{\partial Q_b(\theta, \hat{b}_{s,m})}{\partial f_{MLP}} = \hat{L}(O|\theta) \sum_{t:o_t \in s,m} \Sigma_{s,m}^{-1}(o_t - f_{MLP}(\mu_{s,m})) \quad (5)$$

in which o_t is the current observation vector, $\hat{L}(O|\theta)$ is Viterbi approximation of $L(O|\theta)$ and $\Sigma_{s,m}^{-1}$ refers to the inverse of the covariance matrix of the mth mixture of sth state. The algorithm can be outlined for offline adaptation as follows:

1. Initialize the MLP weights.
2. Read the adaptation utterances and use constrained Viterbi alignment to get the initial segmentation of the utterances. Constrained Viterbi alignment gives the most probable state sequence of the digit string, such that the digits of this string are forced to appear in correct order during recognition.
3. Evaluate the gradient using the Equation (4). Update the weights of the MLP using e.g. backpropagation learning algorithm such that the auxiliary function Q is increased and iterate until Q is maximized [8].
4. Use MLP to transform the means of the HMMs.
5. Repeat steps 2 thru 4 until the maximum number of repetitions is reached or the algorithm has converged.

This algorithm is similar to the generalized expectation maximization (GEM) algorithm used in the original MLP adaptation with the difference that Viterbi approximation is used for the state occupation probabilities[10, 5]. Despite this simplification the algorithm works well in practice which will be shown in Section 4.

3. MLP TRAINING

The structure of MLP was formed by input, hidden and output layers each of which had 39 neurons. The activation functions were chosen to be hyberpolic tangent and linear in the hidden and output layers, respectively. In order to provide a good starting point for MLP training, it was first initialized close to the indentity mapping. This was accomplished such that the $tanh(x)$ activation function was assumed to be almost linear for small x. This gives almost an identity mapping for the hidden layer. When the number of hidden neurons was smaller than the number of inputs, an approximation to the identity mapping was used.

The MLP weights were updated using the resilient backpropagation (RPROP) local adaptive learning scheme at each iteration of the step three of the MLP adaptation algorithm [11, 12]. The training examples for MLP were obtained using the constrained Viterbi alignment of the adaptation utterances as described in Section 2. When there is only a small amount of adaptation data, the training data is a very sparse description of the mapping and can lead to bad generalization of the trained MLP. However, this problem was avoided to some extent by using cross-validation and the weighting of the MLP adapted mean vectors.

The cross-validation set was separated from the adaptation data such that its size was approximately 10% of the original adaptation data set. The training was stopped due to validation failure if the value of Q over the cross-validation set at iteration k was smaller than Q for the iteration $k - 6$. This criterion was found experimentally and it seemed to smoothen out small fluctuations of the Q-function over cross-validation set. The use of cross-validation resulted to approximately one fifth of the training runs ending in validation failure i.e. Q started to decrease. The rest of the runs reached the number of maximum iterations which was set to 150.

After training MLP, the final mean vectors were obtained as a weighted average of the original mean vector and the transformed vector

$$\hat{\mu}_{s,m} = \frac{\alpha \mu_{s,m} + (\beta_{s,m} + 1)\hat{\mu}_{s,m}}{\alpha + \beta_{s,m} + 1} \quad (6)$$

where $\beta_{s,m}$ is the number of the feature vectors assigned to the mth mixture of the sth state. However, it must be emphasized that the weighting of Equation (6) was not used when calculating the value of Q over validation set. This empirical weighting decreased the variance of the test set recognition percentages when having small number of adaptation utterances e.g. less than five adaptation utterances.

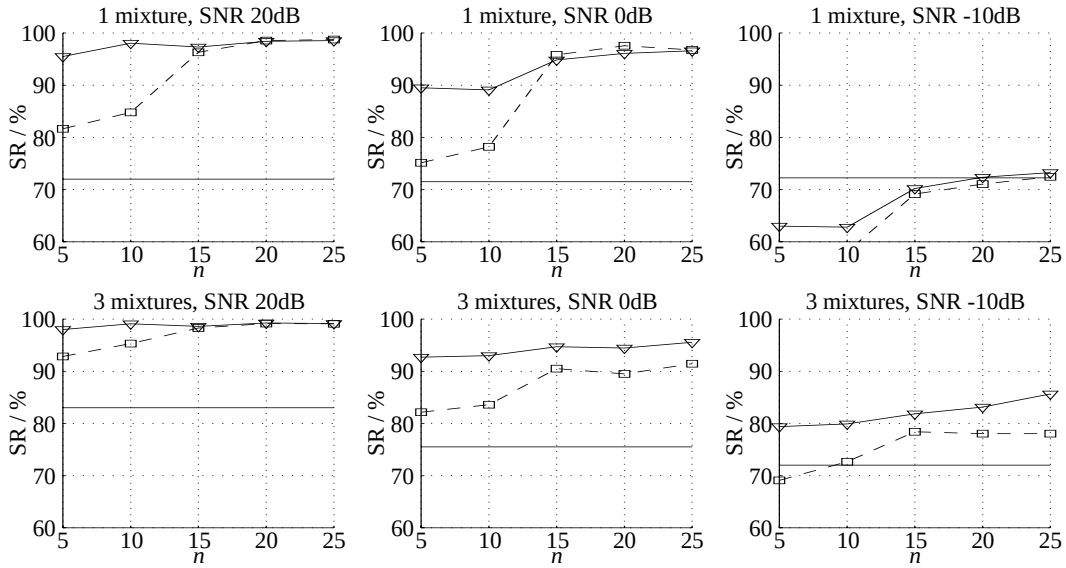

Figure 1: *Average string recognition rates (SR) for three different noise environments when n adaptation utterances with SNR 20dB were used. The solid line with triangles and dashed line with squares refer to the MLP and MLLR adaptation, respectively. The baseline recognition results are shown as straight solid line. The figures in the upper and lower row were obtained using single and 3-mixture models, respectively.*

4. EXPERIMENTAL RESULTS

The MLP and MLLR adaptation methods were tested with adaptation data from five speakers. Each one of the speakers had an adaptation set and a test set which consisted of 400 utterances. The utterances were recorded in three car noise environments with SNRs 20dB, 0dB and -10dB. The number of adaptation utterances varied from 5 to 25 in steps of 5 and only SNR 20dB utterances were used for adaptation. This setting corresponds to the practical usage situation since the users cannot be required to perform speaker adaptation in several noise conditions. The same adaptation utterances were used for both MLP and MLLR adaptation. The number of digits in one utterance varied between three and six in the adaptation and test sets. During recognition, a word loop grammar was used, i.e. in addition to substitution errors, both insertion and deletion errors were possible.

The feature vectors used in the speech recognition system were 39-dimensional with 12 mel-frequency cepstral coefficients, log-energy, and their first- and second-order time derivatives. The tests were run using multi-environment HMMs having one and three mixture components in each HMM state. The digits were modelled with duration constrained left-to-right whole word HMMs.

Figure 1 shows the results of MLP and MLLR adapted recognition systems for the speaker number 5 who has the lowest recognition percentages for the speaker independent baseline speech recognition system. The upper row of Figure 1 has a single mixture models whereas the number of mixtures was three in the lower row. Table 1 summarizes the recognition results for all test speakers with 10 adaptation utterances. Results have been averaged over 5 adaptation experiments. In these results, MLP had one hidden layer with 39 hidden neurons. When having a larger number of hidden neurons, the generalization properties of MLP tend to become worse due to a small amount of adaptation data. Moreover, MLPs with less than 39 hidden neurons could not preserve the mean vectors accurately enough. One iteration of the MLP adaptation algorithm was used during testing.

In Figure 1 it can be seen that MLP adaptation performs better than MLLR when there are a small number of adaptation utterances. Moreover, when having 3-mixture models MLP adaptation achieved better performance also in noisy conditions. This can be also seen in Table 1. Both methods fail in noisy environment if only one mixture is used. This is due to the fact that the initial models have been trained using both clean and noisy speech, but during adaptation only clean speech is available. Hence, the means of the HMMs are tuned to clean speech resulting to enhanced performance in high-SNR conditions and degradation in low-SNR. However, this phenomenon was not disturbing when using 3-mixture models as different noise conditions were more accurately characterized by multiple Gaussian mixtures.

5. CONCLUSION

A nonlinear transformation was presented to transform the mean vectors of continuous density HMMs for speaker adaptation. The mapping was obtained by training an MLP network using the ML criterion to map the mean vectors of speaker independent HMMs to be speaker dependent. The adaptation algorithm was tested in a connected digit speech recognition system under various noise conditions. The obtained results in noise robust

Table 1: *Average performance when 10 clean adaptation utterances were used.*

		SR / %					
		baseline		MLP		MLLR	
Speaker	SNR	1 mix	3 mix	1 mix	3 mix	1 mix	3 mix
1	20dB	99.3	99.0	99.8	99.5	99.6	99.6
	0dB	97.0	99.3	99.3	99.8	94.8	99.6
	-10dB	83.8	94.3	86.3	94.0	81.5	93.0
2	20dB	96.8	99.8	99.4	99.9	97.9	99.8
	0dB	89.8	96.0	97.2	99.0	94.4	98.1
	-10dB	90.3	92.0	90.7	95.0	91.5	94.0
3	20dB	90.8	96.8	98.4	98.8	95.6	98.9
	0dB	91.5	96.0	97.9	98.7	94.3	98.6
	-10dB	75.3	88.3	89.0	93.5	82.4	92.3
4	20dB	83.5	97.0	95.8	98.6	92.2	98.2
	0dB	93.8	99.0	96.1	98.4	89.5	98.0
	-10dB	93.3	94.5	86.5	92.0	78.4	92.0
5	20dB	72.0	83.0	98.1	99.1	84.8	95.3
	0dB	71.5	75.5	89.1	93.0	78.2	83.6
	-10dB	72.3	72.0	62.8	79.9	58.2	72.7
Average	20dB	88.5	95.1	98.3	99.2	94.0	98.4
	0dB	88.7	93.2	95.9	97.7	90.2	95.6
	-10dB	83.0	88.2	83.1	90.8	78.4	88.8

connected digit recognition verify the viability of MLP adaptation. An MLP network can capture the speaker-specific pronounciation details faster than MLLR and it can preserve the multi-environment nature of original HMMs even though adaptation was carried out using only clean speech data. The results show that the proposed method is applicable in speaker adaptation and it provides at least performance comparable to MLLR adaptation.

REFERENCES

[1] C.-H. Lee, C.-H. Lin and B.-H. Juang (Apr. 1991). A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models. *IEEE Transactions on Signal Processing*, 39(4), pp. 806–814.

[2] J.-L. Gauvain and C.-H. Lee (Apr. 1994). Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Transactions on Speech and Audio Processing*, 2(2), pp. 291–298.

[3] L. A. Liporace (Sep. 1982). Maximum Likelihood Estimation for Multivariate Observations of Markov Sources. *IEEE Transactions on Information Theory*, IT-28(5), pp. 729–734.

[4] C. J. Leggetter and P. C. Woodland (Jun. 1994). Speaker Adaptation of HMMs using Linear Regression. Technical Report CUED/F-INFENG/TR.181, Cambridge University Engineering Department.

[5] V. Abrash, A. Sankar, H. Franco and M. Cohen (May 1996). Acoustic Adaptation using Nonlinear Transformations of HMM Parameters. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'96*. pp. 729–732.

[6] M. J. F. Gales and P. C. Woodland (1996). Variance Compensation within the MLLR Framework. Technical Report CUED/F-INFENG/TR.242, Cambridge University Engineering Department.

[7] G. S. Leonard E. Baum, Ted Petrie and N. Weiss (1970). A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41(1), pp. 164–171.

[8] C. M. Bishop (1995). Neural Networks for Pattern Recognition. Oxford University Press, Oxford.

[9] A. J. Viterbi (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, IT-13, pp. 260–267.

[10] A. P. Dempster, N. M. Laird and D. B. Rubin (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1), pp. 1–38.

[11] M. Riedmiller and H. Braun (Mar. 1993). A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: *Proceedings of the IEEE International Conference on Neural Networks ICNN'93*. pp. 586–591.

[12] M. Riedmiller (1994). Advanced Supervised Learning in Multi-layer Perceptrons - From Backpropagation to Adaptive Learning Algorithms. *Computer Standards and Interfaces*, 5. Special Issue on Neural Networks.