TELEPHONE SPEECH RECOGNITION USING NEURAL NETWORKS AND HIDDEN MARKOV MODELS

DongSuk Yuk[†‡] and James Flanagan[†]

[†]CAIP Center, Rutgers University, Piscataway, NJ
[‡]Department of Computer Science, Rutgers University, New Brunswick, NJ
$\{yuk, jlf\}@caip.rutgers.edu$

ABSTRACT

The performance of well-trained speech recognizers using high quality full bandwidth speech data is usually degraded when used in real world environments. In particular, telephone speech recognition is extremely difficult due to the limited bandwidth of transmission channels. In this paper, neural network based adaptation methods are applied to telephone speech recognition and a new unsupervised model adaptation method is proposed. The advantage of the neural network based approach is that the retraining of speech recognizers for telephone speech is avoided. Furthermore, because the multi-layer neural network is able to compute nonlinear functions, it can accommodate for the nonlinear mapping between full bandwidth speech and telephone speech. The new unsupervised model adaptation method does not require transcriptions and can be used with the neural networks. Experimental results on TIMIT/NTIMIT corpora show that the performance of the proposed methods is comparable to that of recognizers retrained on telephone speech.

1. INTRODUCTION

With recent advances in speech recognition technology, continuous density hidden Markov model (HMM) based speech recognizers have achieved a high level of performance in controlled environments, such as matched training and testing environments. However, the recognition performance is typically degraded if the training and testing environments are not matched. One such mismatch is telephone speech recognition using full bandwidth speech recognizers. To achieve good recognition accuracy, speech recognizers are usually retrained using telephone speech data. However, retraining a speech recognizer for telephone speech is an expensive task in terms of training data collection and computation time. In addition, as the recognizer is trained on limited bandwidth speech data, performance is severely degraded compared to high quality speech recognition. Full bandwidth speech recognition usually achieves 90% – 94% word recognition accuracy [14], while telephone speech recognition exhibits only 40% – 65% [11].

In an effort to improve recognition performance, neural networks have been used in conjunction with speech recognizers in various ways for robust speech recognition. In [20], neural networks were applied to reduce noise from noisy speech signals. In [1], a neural network and dynamic time warping (DTW) algorithm were used for speaker dependent word recognition in cars. In [18], two neural networks were used in tandem for both noise reduction and isolated word recognition under F-16 jet noise. In [6], a set of neural networks were used to establish a nonlinear mapping function to transform speech data between two speakers to improve speaker independent recognition performance. In [2], [3], and [15], neural networks were used as a front end of HMM based speech recognizers for feature extraction. In another approach, model adaptation methods for continuous density HMM have been used to best match testing environments by transforming the parameters of speech recognizers (e.g., parameters of Gaussian probability density functions). In maximum *a posteriori* (MAP) based adaptation, existing model parameters are smoothed by new observations and used for speaker adaptation [5]. In parallel model combination (PMC), a clean speech model and a noise model are combined to produce a new noisy speech model for noisy speech recognition [4]. In maximum likelihood linear regression (MLLR), the mean vectors of speaker independent speech recognizers are transformed by affine transformation to best match speaker specific test utterances [9]. In stochastic matching [17], both feature transformation and model transformation are performed using expectation maximization (EM) algorithm. Both [9] and [17] assumed that the relation between training and testing environments is linear.

In this paper, neural network based adaptation methods for robust distant-talking speech recognition [26, 24, 23], which do not require retraining of recognizers, is applied to telephone speech recognition. The feature vector compensation or the model parameter adjustment is automatically learned by the neural networks. Since multi-layer perceptrons (MLP) are known to be able to compute nonlinear mapping functions [10], it can handle the nonlinear distortions found in telephone speech. Secondly, a novel approach for unsupervised model adaptation is proposed. This method does not require transcriptions of adaptation data, by constructing a universal sentence model (USM). It can be used for online adaptation in conjunction with the neural network based adaptation methods. In Section 2, the neural network based adaptation methods are explained. In Section 3, the new unsupervised model adaptation method is proposed. Experimental results are discussed in Section 4.

2. ADAPTATION USING NEURAL NETWORKS

Two types of neural networks that are used for adaptation in robust speech recognition are reviewed in this section; one using a mean squared error (MSE) criterion, and the other using a conditional probability as its objective function.

2.1. Feature Transformation Using Neural Network with Mean Squared Error Criterion

Speech recognizers are trained on wide band speech data. When the testing environment changes, a neural network is trained using a small amount of simultaneously collected speech data (so-called *stereo data*) for the new testing environment. In the case of telephone speech recognition, limited bandwidth speech is provided to the neural network as its input patterns, and the corresponding full bandwidth speech is provided as the target patterns. During the recognition of telephone speech, the neural network transforms input telephone speech feature vectors into those that correspond to high quality speech, and passes them to speech recognizers. An MLP is used to establish the nonlinear mapping function of speech feature vectors between the testing and the training environments. Since the low quality speech feature vectors are transformed to high quality ones, it can outperform the retrained recognizer that is trained on low quality speech. Therefore, the performance upper bound of this approach is not constrained to that of the retrained recognizer [24].

2.2. Maximum Likelihood Neural Network

Neural networks are usually trained to minimize the accumulated MSE [16], E, which is the sum of the squared difference between network output and corresponding target:

$$E = \frac{1}{2}\sum_{o\in O}\sum_{i\in N}(t_{o,i} - o_i)^2 \quad , \qquad (1)$$

where O is observation vectors, N is the number of output nodes (i.e., the dimension of a speech feature vector), o_i is the network output of the ith node for an input vector o (i.e., distorted speech), and $t_{o,i}$ is the corresponding target value (i.e., clean speech). On the other hand, continuous speech recognition is accomplished by finding the word sequence that gives the highest *Viterbi* path likelihood [25]. The acoustic likelihood that is affected by feature transformation is usually computed using a mixture of Gaussian distributions:

$$P(o|q) = \sum_{m\in M} c_m \frac{1}{\sqrt{(2\pi)^N|\Sigma_{q,m}|}} \, e^{-\frac{1}{2}(o-\mu_{q,m})^T\Sigma_{q,m}^{-1}(o-\mu_{q,m})}, \qquad (2)$$

where q is the corresponding state in the Viterbi path, M is the number of Gaussian distributions in the state q, c_m is the weight of mth distribution, and $\mu_{q,m}$ and $\Sigma_{q,m}$ are the mth distribution mean vector and covariances matrix of the state q, respectively. The feature transformation aims to maximize the probability in equation (2). However, minimizing the MSE of neural networks, does not necessarily mean maximizing the acoustic score of equation (2). The anomaly arises from the different criteria, equation (1) and equation (2), in the synergistic use of neural networks and HMM's for robust speech recognition. The maximum likelihood neural network (MLNN) [23] solves this problem of the tandem system by maximizing the likelihood instead of minimizing the MSE. It is known that equation (1) maximizes the likelihood of the correct neural network itself if the target value is distorted by Gaussian noise [13]. However, we do not impose such an assumption, and directly maximize the conditional probability of each state, i.e., equation (2). The error back propagation (EBP) algorithm [16] can still be used with this new objective function. The weight updating rule can be derived by differentiating the logarithm of equation (2) with respect to weight w_{ij} (connection between output node i and hidden node j):

$$\frac{\partial \ln P(o|q)}{\partial w_{ij}} = \frac{\partial \ln P(o|q)}{\partial o_i}\frac{\partial o_i}{\partial w_{ij}} \quad , \qquad (3)$$

where the second term is same as in the original EBP algorithm. The first term is the error at the output layer, and can be rewritten as follows for a diagonal covariances matrix case:

$$\frac{\partial \ln P(o|q)}{\partial o_i} = \frac{1}{P(o|q)}\sum_{m\in M} c_m P(o|q_m)\frac{\mu_{q,m,i} - o_i}{\sigma^2_{q,m,i}} \quad , \qquad (4)$$

where $P(o|q_m)$ is the likelihood of the observation vector o being in the mth distribution of state q, and $\mu_{q,m,i}$ and $\sigma^2_{q,m,i}$ are the ith dimension mean and covariance of the mth distribution in state q, respectively. The error at the output layer is proportional to the weighted sum of the Mahalanobis distance between the mean and the network output. The MLNN is a neural network which takes a distribution instead of a vector, as its target. It should be noted that the MLNN can take any differentiable probability density function, and is not restricted to Gaussian distribution as its target. One advantage of the MLNN is that it does not require stereo data because the target distributions can be obtained using the Viterbi alignment. A similar approach has been used in the context of a maximum likelihood stochastic matching algorithm [19].

The MLNN can be used for model transformation as well as feature transformation when it is used for robust speech recognition. In model transformation MLNN, clean speech model parameters are transformed to distorted speech model parameters to approximate the matched training and testing condition. The new objective function, $P(o|q)$, can also be used for the model transformation. In mean transformation, for example, the observation o is fixed, and the mean $\mu_{q,m}$ is a variable, i.e., network output. Now, the logarithm of equation (2) is differentiated with respect to the network weight w_{ij}:

$$\frac{\partial \ln P(o|q)}{\partial w_{ij}} \;=\; \frac{\partial \ln P(o|q)}{\partial \mu_{q,m,i}}\frac{\partial \mu_{q,m,i}}{\partial w_{ij}} \quad . \qquad (5)$$

The first term of equation (5) can be rewritten as follows for the diagonal covariances matrix case:

$$\frac{\partial \ln P(o|q)}{\partial \mu_{q,m,i}} \;=\; \frac{1}{P(o|q)}\sum_{m\in M} c_m P(o|q_m)\frac{o_i - \mu_{q,m,i}}{\sigma^2_{q,m,i}} \quad . \qquad (6)$$

The weighted difference is propagated from the output layer using the EBP algorithm to best match the mean, $\mu_{q,m,i}$, to its corresponding observation, o_i. A variance transformation MLNN can also be derived in a similar way, where the variable would be $\sigma^2_{q,m,i}$. Unlike feature transformation MLNN, the performance upper bound of model transformation MLNN is constrained to that of retrained recognizers. In general, the mean transformation network can be used where the inverse function may not be physically realizable or where the network can not be well-trained with a limited amount of data.

3. UNSUPERVISED ADAPTATION

In most model adaptation methods such as MAP and MLLR, the transcription of adaptation data is required for training. In order for these methods to be operated in unsupervised mode, the hypothesis from recognition results is usually used as a reference transcription [22]. Instead of using a single hypothesis of the recognized output, multiple *n-best* candidates can be used as the transcriptions [12]. To represent larger number of alternative hypotheses more accurately, *word lattices* can be used instead of a fixed number of hypotheses, in a similar way as [21]. In this study, we propose to construct a universal sentence model (USM) using word HMM's of speech recognizers, and use *bigrams* as the transition probabilities between words[1]. This single HMM can model any utterance (as long as there are no out-of-vocabulary words), and can be used for training without transcription. Since the USM can represent any utterances, it can be considered as a complete word lattice together with language model probabilities. The advantage of this unsupervised adaptation approach is that it can be used together with any other adaptation method discussed so far. Also, any model adaptation algorithm can make use of the USM. The unsupervised adaptation can be used adaptively before recognizing speech in a new environment especially when the environment changes constantly, or incrementally during recognition.

4. EXPERIMENTAL RESULTS

A speech feature vector is composed of 12 dimensional mel-frequency cepstral coefficients (MFCC), normalized energy, and their first and second order time derivatives, resulting in a 39 dimensional vector for 25ms Hamming windowed signals in every 10ms. The baseline speech recognizer is trained using 3,696 utterances from TIMIT training data. It uses 39 phones and 2 silence models. Each phone is modeled using 3-state left-to-right

[1] In the experiment that follows, we did not use bigrams because it was phone recognition experiment.

monophone HMM with 30 Gaussian distributions per state. In total, the system has 3,630 Gaussian distributions. 1,344 utterances from NTIMIT[2] test data are used for testing. When the system is trained and tested under the same environment (i.e., both using TIMIT corpus), the phone recognition accuracy is 62.2% ("TIMIT" in Table 1). When the system is trained using TIMIT and tested using NTIMIT, the accuracy drops to 22.6% ("NTIMIT" in Table 1). The baseline system is retrained using *single pass retraining* algorithm [22] to see the performance of the recognizer trained in the testing environment (i.e., both training and testing use NTIMIT corpus). The performance of the retrained recognizer is 45.4% ("Retrained" in Table 1). However, the retraining requires a large amount of training data (3,696 utterances from NTIMIT training data in this case). For the rest of this paper, the recognizer trained using TIMIT corpus is used for testing, unless stated otherwise.

	sub	del	ins	acc
TIMIT	23.9	9.7	4.1	62.2
NTIMIT	56.1	11.1	10.2	22.6
Retrained	37.2	11.0	6.4	45.4
MSE	43.4	11.9	7.1	37.6
MLNN1	49.6	18.3	7.5	24.6
MLNN1s	48.9	12.0	10.9	28.2
MLNN2	52.6	19.9	4.7	22.8
MLNN2s	48.6	20.3	3.4	27.6
USM	53.3	10.7	9.8	26.2
MLLR1	45.9	16.7	5.5	31.9
MLLR2	45.0	15.6	6.4	33.0
MAP	43.7	13.8	5.8	36.6
ML	44.2	11.2	8.3	36.2
MSE+MLNN1s	47.4	11.6	10.8	30.2
MSE+MLNN2s	48.1	16.9	4.6	30.4
MSE+MLLR1	43.3	13.1	6.6	37.0
MSE+MLLR2	43.2	15.3	5.7	35.8
MSE+USM	42.8	9.0	10.3	38.0
MSE+MAP	41.1	12.6	6.3	39.9
MSE+ML	35.9	7.9	11.3	45.0

Table 1: Phone recognition accuracy in %. "sub", "del", "ins", and "acc" represent substitution error, deletion error, insertion error, and phone recognition accuracy, respectively. "TIMIT" is for matched training and testing using TIMIT corpus. "NTIMIT" is for mismatched training using TIMIT and testing using NTIMIT. "Retrained" is matched training and testing using NTIMIT. The rest are all mismatched conditions (i.e., trained on TIMIT and tested on NTIMIT) with the following adaptation methods. "MSE" is for using the neural network based adaptation method with mean squared error as its objective function. "MLNN1" is the feature transformation maximum likelihood neural network. "MLNN1s" makes use of stereo data in state/frame alignment in addition to "MLNN1". "MLNN2" is the mean transformation maximum likelihood neural network. "MLNN2s" makes use of stereo data as in "MLNN1s". "USM" is unsupervised model transformation using universal sentence model. "MLLR1" is the maximum likelihood linear regression using a single transformation matrix. "MLLR2" is the maximum likelihood linear regression using multiple transformation matrices based on linguistic information. "MAP" is the model transformation using maximum *a posteriori* estimation. "ML" is additional training using *Baum-Welch* algorithm. The rest are the combination of two or more methods in tandem.

The feature transformation neural network with MSE as its objective function performs the best (37.6%) in all the adaptation methods mentioned in the previous section ("MSE" in Table 1). The feature transformation MLNN (24.6%) and the model transformation MLNN (22.8%) improve the performance only marginally ("MLNN1" and "MLNN2" in Table 1, respectively). Particularly, the model transformation MLNN performs poorly compared to feature transformation MLNN. This is probably because only a small number of hidden nodes are used in the model transformation network while the feature transformation MLNN uses a fairly large number of hidden nodes[3], and only the mean vectors are transformed. When stereo data is available, full bandwidth data can be used in state/frame alignment for both the feature transformation MLNN and the model transformation MLNN as in the single pass retraining algorithm. This more accurate state/frame alignment information improves the recognition performance to 28.2% and 27.6% in both cases ("MLNN1s" and "MLNN2s" in Table 1, respectively). The unsupervised model adaptation using USM improves the performance (26.2%) compared to no adaptation (22.6%).

Some of the traditional model transformation methods discussed in Section 1 have been compared. The MLLR that uses a single global transformation[4] does not perform (31.9%, "MLLR1" in Table 1) as well as the one using multiple transformations[5] based on linguistic information (33.0%, "MLLR2" in Table 1). In multiple transformations MLLR, the nonlinearity is approximated as a piece wise linear transformation. However, it does not perform better than feature transformation neural network with MSE (37.6%). The MLLR can be compared to the model transformation MLNN in a sense that both methods transform model parameters in order to best match observed signals. The difference is that MLNN uses nonlinear transformation while MLLR uses (piece wise) linear transformation. In this experiment, it seems that the hidden layer of the MLNN (i.e., VC-dimension [8]) is not complex enough for the amount of adaptation data. The MAP performs better (36.6%) than MLLR (33.0%). This is because MAP uses a larger number of transformations than MLLR, and there is enough adaptation data for those transformations. Additional training using the adaptation data performs (36.2%, "ML" in Table 1) almost same as the MAP.

The neural network approach can be used in combination with the other adaptation methods described above. The feature transformation neural network with MSE can be combined with the feature transformation MLNN (30.2%, "MSE+MLNN1s" in Table 1), with model transformation MLNN (30.4%, "MSE+MLNN2s" in Table 1), or with MLLR (37.0% for "MSE+MLLR1" and 35.8% for "MSE+MLLR2" in Table 1). All these combination degrade original "MSE" performance. On the other hand, when it is combined with the USM (38.0%, "MSE+USM" in Table 1) or the MAP (39.9%, "MSE+MAP" in Table 1), the performance is further improved. The best improvement is achieved by combining the neural network with additional training (45.0%, "MSE+ML" in Table 1). It seems that more fine grain nonlinearity can be captured accurately by ML especially after the feature is transformed by the network. This result is quite comparable to the retrained system (45.4%) which used to be thought of as the upper bound of telephone speech recognition. It should be noted that these adaptation methods use much less training data (462 utterances from training data) than retrained system.

5. CONCLUSIONS

We have described a neural network based transformation approach combined with model transformation methods for robust telephone speech recognition. Experimental results on both the feature transformation and the mean transformation network show 22.1% – 66.4% relative improvement. When the neural network is combined with additional ML training or unsupervised adaptation, the performance improves further, resulting in

[2] NTIMIT is another version of TIMIT recorded via telephone line [7].

[3] The model transformation network uses 39 hidden nodes, and the feature transformation network uses 1000 hidden nodes.

[4] The silence models use seperate transformation.

[5] There are 36 transformations used in this experiment.

a system which is comparable to the retrained recognizer, but with much less training data.

The advantages of this approach are as follows. First, it does not require retraining of the speech recognizer, so the expensive task in terms of training data collection and computational time is avoided. Second, it does not require any knowledge about the distortion, yet it automatically learns the mapping function between training and testing environments. Third, since the MLP is known to be able to compute nonlinear functions, the neural network based approach is able to handle nonlinear distortions found in telephone speech. Finally, the feature transformation neural network using stereo data can learn the inverse distortion function, so its performance upper bound is that of a clean speech recognizer with matched training and testing environments. The mean transformation MLNN does not require stereo data. So, it can be used where the inverse function may not be physically realizable or where the network can not be well-trained with a limited amount of information.

The training algorithm for USM is still open for discussion. Since the competing hypotheses (confusable pairs) can also be represented in a single HMM, discriminative training methods that use *mutual information* criterion instead of *maximum likelihood* may be a good candidate. Currently, only reasonable amount of training data (less than 100 hours of speech) is used to train speech recognizers because the transcriptions have to be made manually. However, using the unsupervised training, it may be possible to make use of the huge amount of untranscribed speech data from TV and radio broadcasting without involving human efforts.

6. ACKNOWLEDGMENT

This research was supported by DARPA contract DABT63-93-C-0037. The authors wish to thank Dr. Samir Chennoukh, Mahesh Krishnamoorthy, Krishna Dayanidhi, and Daniel J. Sinder for their helpful comments and reviewing this paper.

7. REFERENCES

[1] L. Barbier and G. Chollet. Robust speech parameters extraction for word recognition in noise using neural networks. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1:145–148, May 1991.

[2] Y. Bengio, R. Mori, G. Flammia, and R. Kompe. Global optimization of a neural network-hidden Markov model hybrid. IEEE Transactions on Neural Networks, 3(2):252–259, March 1992.

[3] A. Biem and S. Katagiri. Feature extraction based on minimum classification error/generalized probabilistic descent method. IEEE International Conference on Acoustics, Speech, and Signal Processing, 2:275–278, April 1993.

[4] M. Gales and S. Young. Robust continuous speech recognition using parallel model combination. IEEE Transactions on Speech and Audio Processing, 4(5):352–359, September 1996.

[5] J. Gauvain and C. Lee. Maximum *a posteriori* estimation for multivariate Gaussian mixture observations of Markov chains. IEEE Transactions on Speech and Audio Processing, 2(2):291–298, April 1994.

[6] X. Huang. Speaker normalization for speech recognition. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1:465–468, March 1992.

[7] C. Jankowski, A. Kalyanswamy, S. Basson, and J. Spitz. NTIMIT: A phonetically balanced, continuous speech, telephone bandwidth speech database. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1:109–112, April 1990.

[8] M. Kearns and U. Vazirani. An introduction to computational learning theory. MIT Press, 1994.

[9] C. Leggetter and P. Woodland. Flexible speaker adaptation using maximum likelihood linear regression. DARPA Spoken Language Systems Technology Workshop, pages 110–115, January 1995.

[10] R. Lippmann. An introduction to computing with neural nets. IEEE ASSP Magazine, pages 4–22, April 1987.

[11] A. Martin, J. Fiscus, B. Fisher, D. Pallett, and M. Przybocki. 1997 LVCSR/hub-5e whorkshop : Summary of results. DARPA Conversational Speech Recognition Workshop, May 1997.

[12] T. Matsui and S. Furui. N-best-based instantaneous speaker adaptation method for speech recognition. International Conference on Spoken Language Processing, 2:973–976, October 1996.

[13] T. Mitchell. Machine Learning. McGraw-Hill, 1997.

[14] D. Pallett, J. Fiscus, W. Fisher, J. Garofolo, A. Martin, and M. Przybocki. 1995 hub-3 NIST multiple microphone corpus benchmark tests. DARPA Speech Recognition Workshop, February 1996.

[15] M. Rahim and C. Lee. Simultaneous ANN feature and HMM recognizer design using string-based minimum classification error (MCE) training. International Conference on Spoken Language Processing, 3:1824–1827, October 1996.

[16] D. Rumelhart, G. Hinton, and R. Williams. Learning Internal Representations by Error Propagation, D. Rumelhart, J. McClelland, (Eds), Parallel Distributed Processing: Exploration in the Micro-Structure of Cognition, volume 1. MIT Press, 1986.

[17] A. Sankar and C. Lee. A maximum likelihood approach to stochastic matching for robust speech recognition. IEEE Transactions on Speech and Audio Processing, 4(3):190–202, May 1996.

[18] H. Sorensen. A cepstral noise reduction multi-layer neural network. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1:933–936, May 1991.

[19] A. Surendran, C. Lee, and M. Rahim. Unsupervised, smooth training of feed-forward neural networks for mismatch compensation. IEEE Workshop on Automatic Speech Recognition and Understanding, pages 482–489, December 1997.

[20] S. Tamura and A. Waibel. Noise reduction using connectionist models. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1:553–556, April 1988.

[21] V. Valtchev, J. Odell, P. Woodland, and S. Young. MMIE training of large vocabulary recognition systems. Speech Communication, 22(4):303–314, September 1997.

[22] P. Woodland, M. Gales, and D. Pye. Improving environmental robustness in large vocabulary speech recognition. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1:65–68, May 1996.

[23] D. Yuk, C. Che, and J. Flanagan. Robust speech recognition using maximum likelihood neural networks and continuous density hidden Markov models. IEEE Workshop on Automatic Speech Recognition and Understanding, pages 474–481, December 1997.

[24] D. Yuk, C. Che, L. Jin, and Q. Lin. Environment-independent continuous speech recognition using neural networks and hidden Markov models. IEEE International Conference on Acoustics, Speech, and Signal Processing, 6:3358–3361, May 1996.

[25] D. Yuk, C. Che, P. Raghavan, S. Chennoukh, and J. Flanagan. N-best breadth search for large vocabulary continuous speech recognition using a long span language model. 136th meeting of Acoustical Society of America, October 1998.

[26] D. Yuk, Q. Lin, C. Che, L. Jin, and J. Flanagan. Environment-independent continuous speech recognition. IEEE Automatic Speech Recognition Workshop, pages 151–152, December 1995.