CONNECTIONIST SPEAKER NORMALIZATION AND ITS APPLICATIONS TO SPEECH RECOGNITION

X.D. Huang, K.F. Lee, and A. Waibel
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Speaker normalization may have a significant impact on both speaker-adaptive and speaker-independent speech recognition. In this paper, a codeword-dependent neural network (CDNN) is presented for speaker normalization. The network is used as a nonlinear mapping function to transform speech data between two speakers. The mapping function is characterized by two important properties. First, the assembly of mapping functions enhances overall mapping quality. Second, multiple input vectors are used simultaneously in the transformation. This not only makes full use of dynamic information but also alleviates possible errors in the supervision data. Large-vocabulary continuous speech recognition is chosen to study the effect of speaker normalization. Using speaker-dependent semi-continuous hidden Markov models, performance evaluation over 360 testing sentences from new speakers showed that speaker normalization significantly reduced the error rate from 41.9% to 5.0% when only 40 speaker-dependent sentences were used to estimate CDNN parameters.

1. INTRODUCTION

Nonlinear mapping of two different observation spaces is of great interest for both theoretical and practical purposes. In the area of speech processing, nonlinear mapping has been applied to speaker normalization [3, 14, 4, 15, 6], noise enhancement [2, 21], articulatory motion estimation [18, 10], and speech recognition [9]. Recently, nonlinear mapping based on neural networks has attracted considerable attention because of the ability of these networks to optimally adjust the parameters from the training data to approximate the nonlinear relationship between two observed spaces. In speech recognition, speaker variability is one of the major error sources. For example, the error rate of speaker-dependent speech recognition is typically two to three times less than that of speaker-independent speech recognition [7]. To model speaker variability, use of either speaker clustered models [7], or speaker normalization may improve the performance of speaker-independent speech recognition. However, the latter can provide a more compact representation than the former. In addition, speaker normalization can be used to rapidly adapt speaker-dependent models for the new speaker as well as voice conversion for text-to-speech systems [1].

0-7803-0118-8/91/0009-0357$01.00

In this paper, a codeword-dependent neural network (CDNN) is presented for speaker normalization. The network is used as a nonlinear mapping function to transform speech data between two speakers. The mapping function is characterized by two important properties. First, the assembly of mapping functions enhances overall mapping quality. Second, multiple input vectors are used simultaneously in the transformation. This not only makes full use of dynamic information but also alleviates possible errors in the supervision data.

Based on the DARPA Resource Management task [17], large-vocabulary (1000 words) continuous speech recognition was chosen to study the effect of speaker normalization. Speaker-dependent semi-continuous hidden Markov models (SCHMM) [7] were estimated from 2400 sentences (the RM2 training set) [16]. Based on the speaker-dependent SCHMM of one speaker, performance evaluation over 360 testing sentences from the rest of speakers (three speakers) in the RM2 corpus was carried out. Without speaker normalization, the error rate was 41.9% for cross speaker speech recognition. When 40 speaker-dependent adaptation sentences were used, the error rate was reduced to 6.8% based on a single neural network for each new speaker. The CDNN further reduced the error rate from 6.8% to 5.0%. Overall, the error rate was comparable to that of speaker-independent speech recognition on the same testing data.

2. NEURAL NETWORK ARCHITECTURE

2.1. General Principal

Speaker normalization involves acoustic data transformation from one speaker to another. In general, let $\mathcal{X}^a = \mathbf{x}_1^a, \mathbf{x}_2^a, \ldots \mathbf{x}_t^a$ be a sequence of observations at time $1, 2, .. t$ of speaker a. Here, each observation at time k (a frame), $\mathbf{x}_k^a$, is a multidimensional vector, which usually characterizes some short-time spectral features. For speech observations $\mathcal{X}^a$ of speaker a, our goal is to find a mapping function $\mathcal{F}(\mathcal{X}^a)$ such that $\mathcal{F}(\mathcal{X}^a)$ resembles the observation sequences produced by the reference speaker.

Speaker variations include many factors such as sex, vocal tract, pitch, speaking speed, intensity, and cultural differences. Unfortunately, given two different speakers, there is no simple mapping function that can account for all these variations. At a given time t, $\mathbf{x}_t^a$ usually represents some spectral features for the speaker a. In this study, we are mainly concerned with spectral normalization, i.e, to find out a mapping function to transform $\mathbf{x}^a$ so that the normalized observation sequence of speaker a resembles that of the corresponding phonetic realization of speaker b, $\mathbf{x}^b$. Thus, one of the objective functions is to minimize:

$$\sum_{corresponding\ pairs} (\mathcal{F}(\mathbf{x}^a) - \mathbf{x}^b)^2. \qquad (1)$$

Neural networks can be used to approximate any nonlinear mapping function [13]. To be useful for speaker normalization, a layered feedforward neural networks should have a number of features. First, we should have sufficient interconnections between multiple layers and sufficient connections between units in each of these layers so that the mapping network will have the ability to learn complex nonlinear mapping functions between different speakers. Second, as the neural network is suitable only to a small or medium task, the original acoustic space should be partitioned into different prototypes such that each network only performs its own work within the corresponding region.

2.2. Neural Network Topology

It has been found that dynamic information plays an important role in speech recognition [11, 8] As frame to frame normalization lacks use of dynamic information, the architecture of normalization network is thus chosen to incorporate multiple neighboring frames. One such architecture is shown in Figure 1. Here, the current

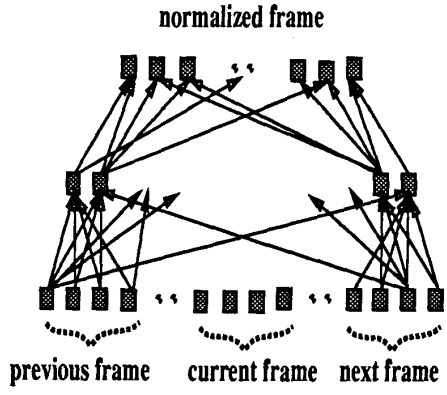

Figure 1: A normalization neural network.

frame and its left and right neighboring frames are fed to the multi-layer neural network as inputs. The network output is a normalized frame corresponding to the current input frame. By using multiple input frames for the network, the important dynamic information can be effectively used in estimating network parameters and in normalization.

If the dimension of observation space is d and the number of input frames is m, we will have dxm input units in the normalization network. This will definitely increase the number of free parameters in the network. Although the increase

in the number of free parameters lead to quick convergence during training, this nevertheless may not lead to improved generalization capability. Since the network is designed to normalize new data from a given speaker to the reference speaker, good generalization capability will be the most important concern.

2.3. Codeword-Dependent Neural Network

When presented with a large amount of training data, a single network is often unable to produce satisfactory results during training as each network is only suitable to a relatively small task. To improve the mapping performance, breaking up a large task and modular construction are usually required [22, 5]. As the nonlinear relationship between two speakers is very complicated, a simple network may not be powerful enough. One solution is to partition the mapping spaces into smaller regions, and to construct a neural network for each region as shown in Figure 2. As each neural

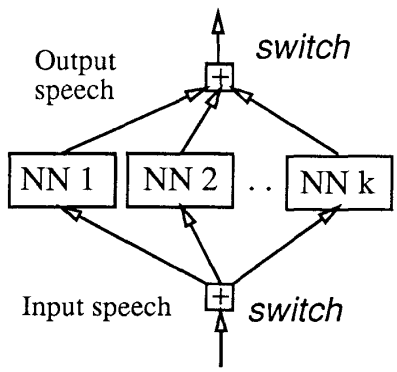

Figure 2: An assembly of neural networks.

network is trained on a separate region in the acoustic space, the complexity of the mapping required of each network is thus reduced. In Figure 2, the switch can be used to select the most likely network or top N networks based on some probability measures of acoustic similarity. Functionally, the assembly of networks is similar to a huge neural network. However, each network in the assembly is learned independently with training data for the corresponding regions. This reduces the complexity of finding a good solution in a huge space of possible network configurations since strong constraints are introduced in performing complex constraint satisfaction in a massively interconnected network.

Vector quantization (VQ) has been widely used for data compression in speech and image processing. Here, it can be used to to partition original acoustic space into different prototypes (codewords). This partition can be regarded as a procedure

to perform broad-acoustic pattern classification. The broad-acoustic patterns are automatically generated via a self-organization procedure based on the LBG algorithm [12].

When the codeword-dependent neural network (CDNN) was constructed from the data in the corresponding cell, it was found that learning for the CDNN converges very quickly in comparison with a huge neural network. The larger the codebook, the quicker it converges. However, the size of codebook relies on the number of available training data since codeword-dependent structure fragments training data. The size of codebook should be determined experimentally.

2.4. Modifications to Sigmoid Functions

The basic unit used in many neural networks computes the weighted sum of its inputs and passes this sum through a nonlinear function such as a threshold or sigmoid function [19] as:

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

In the standard sigmoid function, the output dynamic range is between 0 and 1. If the mapping function input and output are out of this range, feature conditioning is generally required. However, if speaker normalization is applied to speaker adaptation, the original acoustic data can not be conditioned. Because of this, all the output units in the network are not associated with any sigmoid function. Instead, the linear function is used. In addition, the sigmoid function is generalized as:

$$SIGMOID(x) = \frac{\alpha}{1 + e^{-\beta * x}} - \gamma \tag{3}$$

Using $SIGMOID$ function, the dynamic range and the shape can be easily controlled according to the observation structure. In fact, these parameters can also be learned automatically during backpropagation.

3. SPEAKER NORMALIZATION EXPERIMENTS

We want to see if speaker normalization can minimize speaker variations with only a limited amount of training sentences. Consequently, we used 40 speaker-dependent sentences (about 5 minutes) to estimate the network parameters to map data of new speakers to the reference speaker.

The DARPA's resource management task [17] is used for the development of speaker-dependent models, which consists of 2400 training sentences from one male speakers (LPN) and 120 testing sentences (RM1, used in June 1990 evaluations)

from 3 new speakers. The speaker-dependent training set includes 99% of the words in the vocabulary. The testing set includes 73% of the words in the vocabulary. For neural network training, 40 sentences are randomly extracted from the speaker-dependent training set. The word coverage is less than 19% in the normalization training set. Both testing and training have the same recording conditions. A number of experiments have been conducted on these data sets. The reader is referred to [7] for a summary of some recognition performance benchmarks.

Through this study, the feature extraction performed in both training and testing is a LPC-based cepstral coefficients with a 20-ms Hamming window and a 10-ms frame shift. Bilinear transformation of cepstral coefficients is employed to approximate mel-scale representation. Multiple-string features including first-order and second-order time derivatives are used to construct four independent codebooks [7]. Each codeword is modeled by a Gaussian density function. The top-4 codewords are used for the semi-continuous output probability density function. Only the cepstral vectors are considered for normalization. Once we have the normalized cepstral vector, the first-order and second-order time derivatives can be computed.

3.1. Benchmark experiments

As benchmark experiments, speaker-dependent speech recognition was first evaluated. The training set consists of 2400 sentences from each speaker. We used generalized triphone models derived from speaker-independent training set [11]. Based on speaker-dependent data, the SCHMM parameters and VQ codebook were estimated jointly starting with sex-dependent models and codebooks. The average error rate for four speakers was 1.4% as shown in Table 1. Here, the error rate of the

Speaker	2400 Training Sentences Word Error Rate
BJW	1.0%
JLS	2.7%
JRM	1.5%
LPN	0.4%
Average	1.4%

Table 1: Speaker-dependent results.

speaker-dependent system is about three times less that of the speaker-independent system [7]. When we used one male speaker (LPN) as the reference speaker, and designated the other three (one male and two female) as testing speakers, the word recognition error rate was 41.9% as shown in Table 2. We can see that the error rate of the female speakers (BJW and JRM) increased substantially.

Speaker	Cross speaker recognition Word Error Rate
BJW	55.3%
JLS	8.5%
JRM	62.1%
Average	41.9%

Table 2: Cross speaker recognition results.

3.2. Normalization Results

To provide learning examples for the network, a DTW algorithm [20] was used to warp the target data to the reference data. For the given input frames, the desired output frame for network learning is the one paired by the middle input frame in DTW alignment. One important caution in applying the DTW alignment is that silence should be excluded.

The input of the network consists of three frames from the target speaker. Here, 12 cepstral coefficients and energy are used together. Thus, there are 93 input units in the network. The output of the network has 13 units corresponding the normalized frame, which is made to approximate the frame of the desired reference speaker. The energy output is discarded as it is relative unstable. The objective function for network learning is to minimize the distortion (mean squared error) between the network output and the desired reference speaker frame. The network has one hidden layer with 20 hidden units. Each hidden unit is associated with the $SIGMOID$ function, where α, β and γ are predefined to be 4.0, 1.8, 2.0 respectively. They are fixed for all the experiments conducted here. Experimental experience indicates that 300 to 600 epochs are required to achieve acceptable distortion. We averaged results of three runs in following experiments.

When a single network was used for each speaker, the average word error rate was reduced from 41.9% to 6.8% as shown in Table 3 (VQ size = 1). Although neither the codebook nor the HMM parameter was adapted in this experiment, the error rate was already reduced by a factor of 6. It is also interesting to note that for female speakers (JRM and BJW), speaker normalization dramatically reduces the error rate.

To improve the generalization capability, one can increase the number of training data or reduce the number of free parameters. However, the complexity or dimensionality of the mapping network usually has to be increased to maintain accurate mapping between two speakers, which leads to the increased effective number of degrees of freedom in the networks. Therefore, it is important to smooth the less-well trained parameters. The nonlinear network output can be interpolated with the

original network inputs [6]. A linear input-output feedforward path can added to the network. All the input units can be either fully or partially connected to the output units without passing through any nonlinear function. The rational to add an interpolation path is that the nonlinear network can not be well constructed. Such a network architecture interpolates the nonlinear network output with the original network input with the interpolation weights automatically determined by the error back propagation algorithm [19]. However, this topology only works well for some speakers. There is no significant overall difference in comparison with the basic topology.

When the CDNN was used, we observed additional 25% error reduction. The error rate was further reduced from 6.8% to 5.0% as shown in Figure 3. This error rate is comparable to that of the best speaker-independent performance on the same test set [7]. This indicates the assembly of mapping functions indeed enhances the overall mapping quality. The best performance was attained when the codebook size was between 4 to 8. Further increase in the codebook size, as shown in the figure, led to degraded performance because of too many free parameters.

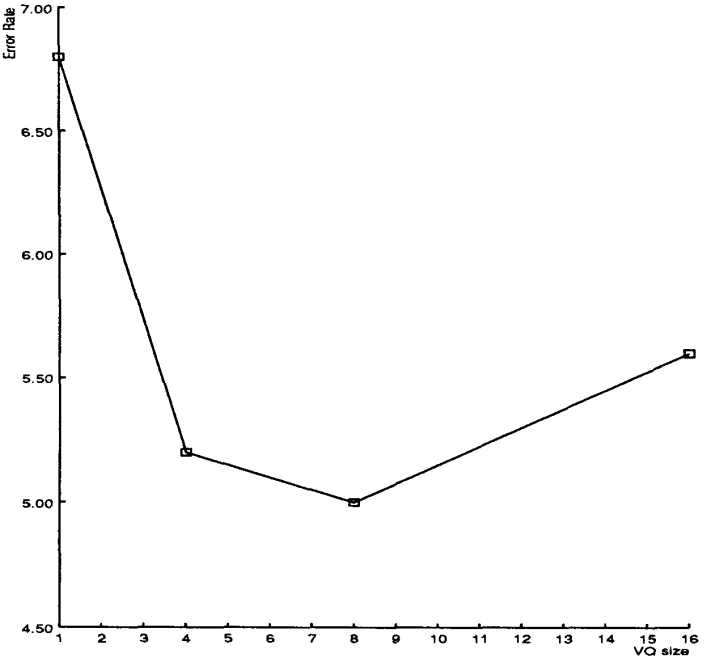

Figure 3: Codeword-dependent network performance.

4. SUMMARY

In this paper, the codeword-dependent neural network (CDNN) is presented for speaker normalization. The network is used as a nonlinear mapping function to transform speech data between speakers. Large-vocabulary continuous speech recognition is chosen to study the effect of speaker normalization. Using speaker-dependent SCHMMs on the DARPA RM task, performance evaluation from three new speakers showed that speaker normalization reduced the error rate from 41.9% to 5.0% when only 40 speaker-dependent sentences are used to estimate CDNN parameters. The error rate is comparable to that of the best speaker-independent performance. Our mapping function is characterized by two important properties. First, the assembly of mapping functions enhances the overall mapping quality. Second, multiple input vectors are used simultaneously in normalization.

Speaker-independent network can also be used as part of the front-end of the speaker-independent speech recognition system. The network can be built to reduce the difference among speakers before speaker-independent HMM training is involved such that speaker-independent models will have sharper distributions (better discrimination capability) in comparison with the conventional training procedure. Use of such normalization networks for speaker-independent speech recognition as well as unification of the SCHMM and speaker normalization could provide a new computational architecture for speech recognition.

Acknowledgments

This research was sponsored by the Defense Advanced Research Projects Agency (DOD), Arpa Order No. 5167, under contract number N00039-85-C-0163. The authors would like to express their gratitude to Professor Raj Reddy for his encouragement and support, and other members of CMU speech group for their help.

References

[1] Abe, M., Nakamura, S., and Shikano, K. *Voice conversion through vector quantization.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing.** 1988.

[2] Acero, A. and Stern, R. *Environmental Robustness in Automatic Speech Recognition.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing.** 1990, pp. 849–852.

[3] Choukri, K., Chollet, G., and Grenier, Y. *Spectral transformations through cannonical correlation analysis for speaker adapataion in ASR.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing.** 1986, pp. 2659–2552.

[4] Class, F., Kaltenmeier, A., Regel, P., and Trottler, K. *Fast speaker adaptation for speech recognition.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing.** 1990, pp. 133–136.

[5] Hampshire, J. and Waibel, A. *The Meta-Pi Network: Connectionist rapid adapatation for high-performance multi-speaker phoneme recognition.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing.** 1990, pp. 165–168.

[6] Huang, X. *A Study on Speaker-Adaptive Speech Recognition.* in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[7] Huang, X. and Lee, K. *On Speaker-Independent, Speaker-Dependent, and Speaker-Adaptive Speech Recognition.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1991, pp. 877–880.

[8] Huang, X., Lee, K., Hon, H., and Hwang, M. *Improved Acoustic Modeling for the SPHINX Speech Recognition System.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1991, pp. 345–348.

[9] Iso, K. and Watanabe, T. *Speaker-independnet word recognition using a neural prediction model.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1990, pp. 441–444.

[10] Kobayashi, T., Yagyu, M., and Shirai, K. *Applications of neural networks to articulatory motion estimation.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1991, pp. 489–4920.

[11] Lee, K., Hon, H., and Reddy, R. *An Overview of the SPHINX Speech Recognition System.* **IEEE Transactions on Acoustics, Speech, and Signal Processing**, January 1990, pp. 35–45.

[12] Linde, Y., Buzo, A., and Gray, R. *An Algorithm for Vector Quantizer Design.* **IEEE Transactions on Communication**, vol. COM-28 (1980), pp. 84–95.

[13] Lippmann, R. *Neural Nets for Computing.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1988, pp. 1–6.

[14] Montacie, C., Choukri, K., and Chollet, G. *Speech recognition using temporal decomposition and multi-layer feed-forward automata.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1989, pp. 409–412.

[15] Nakamura, S. and Shikano, K. *A comparative study of spectral mapping for speaker adaptation.* **ICASSP**, 1990, pp. 157–160.

[16] Pallett, D., Fiscus, J., and Garofolo, J. *DARPA Resource Management Benchmark Test Results June 1990.* in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1990, pp. 298–305.

[17] Price, P., Fisher, W., Bernstein, J., and Pallett, D. *A Database for Continuous Speech Recognition in a 1000-Word Domain.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1988, pp. 651–654.

[18] Rahim, M., Kleijn, W., Schroeter, J., and Goodyear, C. *Acoustic to articulatory parameter mapping using an assembly of neural networks.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1991, pp. 485–488.

[19] Rumelhart, D., Hinton, G., and Williams, R. *Learning Internal Representation by Error Propagation.* in: **Learning Internal Representation by Error Propagation**, by D. Rumelhart, G. Hinton, and R. Williams, edited by D. Rumelhart and J. McClelland. MIT Press, Cambridge, MA, 1986.

[20] Sakoe, H. and Chiba, S. *Dynamic Programming Algorithm Optimization for Spoken Word Recognition.* **IEEE Transactions on Acoustics, Speech, and Signal Processing**, vol. ASSP-26 (1978), pp. 43–49.

[21] Tamura, S. and Waibel, A. *Noise reduction using connectionist modelsnce Measure for Speech Recognition.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1988, pp. 553–556.

[22] Waibel, A., Sawai, H., and Shikano, K. *Consonant Recognition by Modular Construction of Large Phonemic Time Delay Neural Networks.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1989.