MAXIMUM-LIKELIHOOD STOCHASTIC MATCHING APPROACH TO NON-LINEAR EQUALIZATION FOR ROBUST SPEECH RECOGNITION

A. C. Surendran [*†], *Chin-Hui Lee* [††] *and Mazin Rahim* [†††]

[†]Rutgers University, [††]Bell Laboratories,[†††] AT&T Laboratories

ABSTRACT

In this paper we present a new technique in the stochastic matching framework to compensate for non-linear distortions in speech recognition. The features of the test data and the means of the trained model are both transformed using neural networks to better fit each other. The parameters of the neural network are estimated using a novel combination of the generalized EM (GEM) and the backpropagation algorithms. In the feature transformation case, when the exact Q-functions cannot be calculated, approximations are heuristically derived. The mathematical properties of the new algorithm are analyzed. The performance of the algorithm is also studied under different mismatch conditions.

1. INTRODUCTION

Acoustic mismatches between the training and the testing conditions often lead to degradation in the performance of automatic speech recognizers. These mismatches may be due to speaker variation, change in the transducer, channel effects, competing noise sources, or an unknown combination of the above. In general, they affect the features of speech in a non-linear fashion which is not easily characterized. Most approaches either adapt the features [1, 2, 3] or the model parameters [4, 5]. Recently, *stochastic matching*, a framework for adapting both the features and the models using the maximum likelihood approach was developed [6].

As shown in Figure 1, given the test utterance and the model pair, $(\mathbf{Y}, \Lambda_X)$, the stochastic matching approach reduced the mismatch in two ways: (1) By transforming the test features $\mathbf{X} = F_\nu(\mathbf{Y})$; and (2) by transforming the model $\Lambda_Y = G_\eta(\Lambda_X)$, where ν and η are the parameters of the transformation.

The parameters are estimated such that the likelihood of the data is maximized, i.e.,

$$\nu = \underset{\nu}{\operatorname{argmax}}\, P(\mathbf{Y}|\nu, \Lambda_X). \tag{1}$$

In simple cases, this maximization can be done using the EM algorithm [6, 7] in which an auxiliary or a Q-function

$$Q(\nu'|\nu) = E\{\log P(\mathbf{Y}|\nu', \Lambda_X)|\nu, \Lambda_X\}, \tag{2}$$

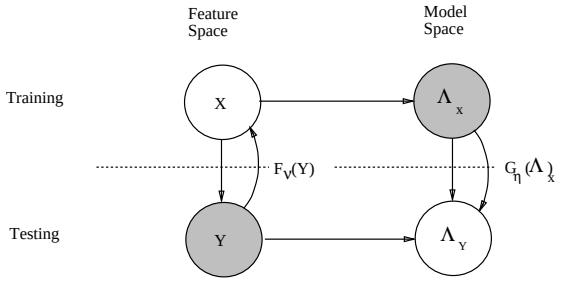

Figure 1. Stochastic matching using parametric transformations of both features and models

is maximized which leads to a maximization of the likelihood.

Earlier approach in this framework was primarily in compensating for linear distortions [6] in the cepstral domain. In this paper we propose a new technique that extends this approach to compensate for non-linear distortions. To achieve an optimal performance, the functional forms of F_ν and G_η can be selected based on the *a priori* knowledge of the degrading mechanism. But in most applications, it is not possible to easily characterize the degradation or quantify it mathematically. Hence in this paper, we assume a data-driven approach. F_ν and G_η are modeled using multilayer perceptrons (MLP). In this case, a direct maximization of the Q-function is not possible. We propose a technique which estimates the weights using a novel combination of the backpropagation algorithm and the generalized EM (GEM) algorithm [7, 8].

2. NON-LINEAR TRANSFORMATION OF MODEL PARAMETERS

In [6], the stochastic matching approach was used to estimate additive corrections to model means and variances without the need to retrain the models. A similar approach was used to estimate a linear transformation of model means for speaker adaptation based on a set of training data [4]. Here, we extend this approach to transform the model means using a non-linear transformation. Thus

$$\mu_Y = G_\eta(\mu_X), \tag{3}$$

where μ_Y is the mean of the test data, μ_X is the mean of the trained models, and G_η is an artificial neural network (ANN).

*This research was conducted at AT&T Bell Laboratories as a part of A. C. Surendran's Ph.D. thesis

The E-step of the EM algorithm can be applied directly to estimate the implied auxiliary function [9]

$$Q(\eta'|\eta) = E\{\log P(\mathbf{Y}|\eta', \Lambda_X)|\eta, \Lambda_X\}. \quad (4)$$

But the M-step cannot be applied directly since a closed form solution does not exist. In this case, it can be replaced by a generalized M-step [7, 9] where at each step i, η^{i+1} can be estimated such that

$$Q(\eta^{i+1}|\eta^i) \geq Q(\eta^i)|\eta^i), \quad (5)$$

where η^i is the parameter set at the i^{th} step. This guarantees that the likelihood never decreases. The challenge is to use this step to estimate the parameters of the multilayer perceptron. Traditionally, the MLP weights are trained by updating them in the direction of the gradient of an error metric using some target values. Here, we modify the approach to use a different objective function - one that does not need target values, but requires only the inputs and the models of the target; specifically, we adapt the MLP weights in the direction of the gradient of $Q(\eta^i|\eta^i)$. This ensures that the Q-function increases. Thus the new parameter set is

$$\eta^{i+1} = \eta^i + \alpha \frac{\partial Q(\eta|\eta^i)}{\partial \eta}|_{\eta=\eta^i}, \quad (6)$$

where α is the step size, which chosen to ensure smooth convergence.

For a Hidden Markov Model (HMM) with N states and M Gaussian mixtures per states, the Q-function can be written as [9]

$$Q(\eta'|\eta) = \sum_{t=1}^{T}\sum_{n=1}^{N}\sum_{m=1}^{M}\gamma_t(n,m)\cdot\left[\sum_{j=1}^{D}\frac{(y_{t,j} - G_{\eta',j}(\mu_{n,m}))^2}{\sigma^2_{n,m,j}}\right], \quad (7)$$

where $\gamma_t(n,m) = P(\mathbf{y}_t, n, m|\eta, \Lambda_X)$ is the joint probability of the observations being in state n mixture m at time t. Here we assume that the covariance matrices are diagonal, which makes it possible to decompose the neural network into one for each component of the mean.

Using the backpropagation algorithm, different update rules can be derived for updating the weights of the outer layer and the hidden layers [9]. The transformation can be segmental. i.e., parameters in "similar" states can be tied to reduce the number of parameter updates and to improve generalization using the test data.

3. FEATURE TRANSFORMATION FOR NON-LINEAR DISTORTIONS

In the case of the feature transformation, if each frame of the test frame is transformed into an estimate of the training data,

$$\mathbf{x}_t = F_\nu(\mathbf{y}_t), \quad (8)$$

the density function of $\mathbf{y}_t$, $P(\mathbf{y}_t, n, m|\Lambda_X)$ for mixture m in state n can be written in terms of the corresponding density of $\mathbf{x}_t$, $P(\mathbf{x}_t, n, m|\Lambda_X)$, and the Jacobian of the transformation $J = \frac{\partial \mathbf{y}_t}{\partial F_\nu(\mathbf{y}_t)}$. The Q-function can be calculated for simple transformations and the parameter ν can be estimated by

$$\nu = \underset{\nu'}{\text{argmax}}\, E\{\log P(\mathbf{Y}|\nu', \Lambda_X)|\nu, \Lambda_X\}. \quad (9)$$

3.1. Alternate Choices of Q-functions

For more complicated forms of F_ν, such as neural networks, it is often not possible to calculate the Jacobian and the density function of $\mathbf{y}_t$. In such cases, instead of maximizing the likelihood $P(\mathbf{Y}|\nu', \Lambda_X)$, the likelihood of the transformed data $\hat{\mathbf{X}}$, $P(\hat{\mathbf{X}}|\nu', \Lambda_X)$ can be maximized. The effectiveness of the approximate Q-function can be tested by using it to estimate an *affine transformation* $\mathbf{y}_t = A\mathbf{x}_t + b$. The expressions for estimating A and b are given in [6]. Similar expressions can be derived using the approximate Q-function [9].

A test experiment was performed where the data under the matched condition was artificially degraded using a known affine transformation. In this experiment, A was assumed to be a diagonal matrix. Thus the test data were obtained from the training data as $y_{t,i} = \frac{x_{t,i} - b_i}{a_i}$, $i = 1, \cdots, D$, where $\{a_i, b_i\}$ are predetermined values. Now, these values were calculated using the exact (equation 9) and approximate Q-function mentioned above (denoted by Q1 and Q2 respectively). The results are given in Table 1. The values are given only for the first five coefficients.

Coeff	Original weights		Wts from Q1		Wts from Q2	
	a	b	a	b	a	b
1	1.25	0.25	1.49	0.28	1.05	0.24
2	1.2	-0.2	1.4	-0.22	0.86	-0.180
3	1.0	0.15	1.4	0.013	0.71	0.137
4	0.9	0.1	0.95	0.044	0.6	0.01
5	1.1	0.1	1.38	0.014	0.79	0.07

Table 1. Weights calculated for the affine transform using Q1 and Q2

The first Q function consistently over estimated the weights and the second Q function consistently underestimated the weights. This was a motivation to heuristically derive a "Joint Q-function" (JQ) that is a combination of the two Q-functions. The resulting network estimates are given in Table 2

Coeff	Original weights		Weights from JQ	
	a	b	a	b
1	1.25	0.25	1.23	0.25
2	1.2	-0.2	1.16	-0.20
3	1.0	0.15	1.07	0.09
4	0.9	0.1	0.82	0.05
5	1.1	0.1	1.10	0.04

Table 2. Weights calculated for the affine transform using the Joint Q-function.

It is clear that the weights estimated from the Joint-Q function are closer to the actual values, compared to Q1 or Q2. The corresponding recognition results are given in

Table 3. The Joint-Q function in this case restores the lost performance whereas Q1 does slightly worse. Q2 does not compete with Q1 or the Joint-Q function. This performance improvement of the joint-Q function is not uniform across all experiments. The reasons for this will be further explored in the next section.

Condition	Word Error(%)
Matched	2.7%
Degraded	3.6%
Q1	3.1%
Q2	8.1%
Joint Q-function	2.7%

Table 3. Word error rate when degrading speech using an affine transformation and adapting using Q1, Q2 and Joint Q.

Using the approximate Q-function and ANNs to transform the features, equations can be derived for adapting the weights using the GEM algorithm, just as in the case of the model transformation [9]. Preliminary experiments suggest that using Q2 with neural networks is not useful directly for improving recognition performance.

4. RESULTS

A stereo database, similar to the one used in [6], was used in our experiment. Sentences from the 991-word DARPA resource management (RM) task were recorded simultaneously through two channels: (1) A close talking microphone (MIC), and (2) a telephone handset over a dial-up line (TEL). The sentences were spoken by a non-native (NN) speaker. The data consisted of 300 sentences for training and 75 sentences for testing. 1769 context dependent (CD) subword unit models were built, with a maximum of 16 mixtures per state. The RM word pair grammar which gives a perplexity of about 60 was used for the experiments. Starting from a set of speaker independent HMMs (Λ_{SI}) and using the 300 sentences recorded over the MIC channel a data-specific model (Λ_{MIC}) was generated using MAP adaptation [5]. The baseline performances are given in Table 4.

	Λ_{SI}	Λ_{MIC}
NN-MIC	24.0	2.1
NN-TEL	63.3	20.1

Table 4. Baseline performance in word error (%)

Recognition results are provided for three different cases, each of which demonstrates the effect of the algorithm under different mismatch conditions:

- Λ_{SI}, the speaker-independent model used with NN-MIC data - speaker mismatch;
- Λ_{SI}, the speaker-independent model used with NN-TEL data - channel, speaker and transducer mismatch with possible additive noise; and
- Λ_{MIC}, the speaker-adaptive MIC model used with NN-TEL data - channel and transducer mismatch.

The knowledge that the databases were recorded simultaneously was never used in the adaptation technique.

The recognition results for the three cases are given in Tables 5,6 and 7 respectively . 'B' indicates estimation of a bias in the feature domain using the gradient computation; 'Q1' indicates the direct calculation of an affine transform using the exact Q-function; 'JQ' indicates the calculation of the affine transform using the joint Q-function. In the model transformation domain, a single layer ANN is equivalent to a linear transform of the model means. Thus 'MD' denotes a directly computed linear regression transform and 'MG (linear)' denotes calculation using the gradient approach. The model transformation is done in three different ways: (1) using a single layer ANN; (2) using an ANN with one hidden layer of 3 neurons (denoted by 'MG (non-linear)'), and (3) using a single layer ANN followed by a ANN with a hidden layer (denoted by 'MD (lin+nl)'). The digit following each symbol denotes the number of transforms used i.e., 'B-1' denotes that one transform was used for all features; 'B-2' denotes that two transforms were used, one for speech and one for silence. 'B-6' denotes that 6 transforms, one for each phonemic class was used. In the case of 1 and 2 transforms, only one sentence was used for adaptation; when using 6 transforms, to avoid the problem of limited data, 15 sentences were used for adaptation.

Λ_{SI} models and NN-MIC data	
Transformation Used	Word Error Rates (%)
Baseline	24.0
B-1	19.8
B-2	19.1
B-6	20.4
Q1-1	21.9
Q1-2	19.8
Q1-6	23.1
JQ-1	24.2
JQ-2	20.7
JQ-6	23.7
MD-6	18.5
MG-6 (linear)	15.2
MG-6 (non-linear)	15.7
MG-6 (lin+nl)	16.3

Table 5. Word error rate (%) using Λ_{SI} models and testing on MIC data.

The affine transformation of the test data models additive noise in addition to linear channel effects. From the results of the''B','Q1' and 'JQ' experiments from Tables 5, 6 and 7, it can be seen that the affine transform gives improvement in performance over the bias case only in the presence of additive noise i.e., when the data is recorded over the telephone channel. The joint Q-function performs better at lower degradation levels (Table 7). Its performance drops behind that of the original Q-function as the degradation becomes more severe (Table 6). The approximate Q-function that maximizes the likelihood of $\dot{\mathbf{X}}$ (Q1) did not perform well and hence the results are not reported here. This may be because the maximization of $P(\dot{\mathbf{X}}|\nu, \Lambda_X)$

Λ_{SI} models and NN-TEL data	
Transformation Used	Word Error Rates
Baseline	63.3
B-1	54.1
B-2	53.8
B-6	53.2
Q1-1	41.1
Q1-2	51.7
Q1-6	46.4
JQ-1	49.1
JQ-2	47.7
JQ-6	46.8
MD-6	31.6
MG-6 (linear)	30.2
MG-6 (non-linear)	39.1
MG-6 (lin+nl)	27.2

Table 6. Word error rates (%) using Λ_{SI} models and testing on TEL data.

Λ_{MIC} models and NN-TEL data	
Transformation Used	Word Error Rate
Baseline	20.1
B-1	12.2
B-2	12.0
B-6	10.5
Q1-1	11.1
Q1-2	11.7
Q1-6	14.3
JQ-1	9.8
JQ-2	10.1
JQ-6	9.2
MG-6 (linear)	5.7
MG-6 (lin+nl)	5.7
Matched	2.7

Table 7. Word error rates (%) using Λ_{MIC} models and testing on TEL data

may allow for unrestricted movement of the weights as long as the likelihood increases. One way to make it perform better is to define bounds that restrict the movement of the cepstra. This may also explain the behavior of the joint Q-function. At low levels of degradation, the addition of Q2 to Q1 might reinforce the segmental constraints on $\mathbf{Y}$. At higher levels of degradation, Q2 may become less relevant.

The 'MD' and 'MG' results of Tables 5, 6 and 7 show that the GEM using the linear ANN performs comparably to the direct linear transform. They also show that this approach can successfully alleviate both channel and speaker mismatches . Neural networks with hidden layers seem to model the additional non-linear effects due to the interaction of speaker, channel and noise. But they only provide a slight improvement over the linear case. This can be seen from the fact that it gives performance improvement only when used with Λ_{SI} and TEL data (Table 6). Gradient descent gives algorithmic problems when using Q1 or JQ in non-linear compensation cases [9]. This has to be studied more carefully.

5. SUMMARY

In this paper we have introduced a technique that can potentially compensate for non-linear distortions in speech recognition using ANNs in the stochastic matching framework. Model means and (potentially) features of the test data, can be transformed using ANNs which model non-linear transforms. We derived an algorithm that uses the generalized EM algorithm and the backpropagation algorithm to adapt the neural network weights. The earlier linear approaches can be developed as special cases under this approach. Experimental results show that the technique is helpful under various mismatch conditions, though the non-linear transform shows only a modest improvement over the linear approaches in most cases. In the feature domain, we developed an approximate Q-functions instead of the incalculable exact Q-function. Though this was not directly beneficial, it was useful through a joint Q-function in some cases. Further study is needed to formulate meaningful Q-functions that are also mathematically tractable.

REFERENCES

[1] Rahim, M. and Juang, B.H., "Signal Bias Removal for Robust Telephone Speech Recognition in Adverse Environments", Proc. ICASSP, 1994.

[2] Zhao, Y., "A New Speaker Adaptation Technique Using Very Short Calibration Speech", Proc. ICASSP, Vol. II, pp. 562-565 , 1993.

[3] Furui, S., "Unsupervised Speaker Adaptation Based on Hierarchical Spectral Clustering", IEEE Trans. ASSP, Vol. 27, 1923-1930.

[4] Leggetter, C.J. and Woodland, P. C., "Speaker Adaptation of HMMs Using Linear Regression", Cambridge University Engineering Department TR-181, 1994.

[5] Lee, C.-H., and Gauvain, J.-L., "Speaker Adaptation based on MAP Estimation of HMM Parameters", ICASSP, pp. II-558-561, 1993.

[6] Sankar, A. and Lee., C.H., "A Maximum Likelihood Approach to Stochastic Matching for Robust Speech Recognition", IEEE Trans. on SAP, 1995.

[7] Dempster, A.P., Laird, N.M. and Rubin, D.B., "Maximum Likelihood from Incomplete Data via the EM Algorithm", J. Royal Stat. Soc. (B), Vol. 39, pp.1-38.

[8] Wu, C.F.J., "On the Convergence Properties of the EM Algorithm", Ann. Stat., Vol. 11, No. 1, pp.95-103, 1983.

[9] Surendran, A. C., "Maximum-likelihood Stochastic Matching Approach to Non-linear Equalization for Robust Speech Recognition", Ph.D. Thesis, Rutgers University, 1996.