NOISE ADAPTIVE TRAINING USING A VECTOR TAYLOR SERIES APPROACH FOR NOISE ROBUST AUTOMATIC SPEECH RECOGNITION

Ozlem Kalinli, Michael L. Seltzer, and Alex Acero*

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052
kalinli@usc.edu, {mseltzer, alexac}@microsoft.com

ABSTRACT

In traditional methods for noise robust automatic speech recognition, the acoustic models are typically trained using clean speech or using multi-condition data that is processed by the same feature enhancement algorithm expected to be used in decoding. In this paper, we propose a noise adaptive training (NAT) algorithm that can be applied to all training data that normalizes the environmental distortion as part of the model training. In contrast to the feature enhancement methods, NAT estimates the underlying "pseudo-clean" model parameters directly without relying on point estimates of the clean speech features as an intermediate step. The pseudo-clean model parameters learned with NAT are later used with vector Taylor series (VTS) model adaptation for decoding noisy utterances at test time. Experiments performed on the Aurora 2 and Aurora 3 tasks, demonstrate that the proposed NAT method obtain relative improvements of 18.83% and 32.02%, respectively, over VTS model adaptation.

Index Terms— Noise adaptive training, model adaptation, robust automatic speech recognition, vector Taylor series.

1. INTRODUCTION

The performance of automatic speech recognition (ASR) in noisy environments has not yet reached a desired level of performance despite years of research. It remains a challenging problem since there are many possible types of environmental distortion, and it is difficult to compensate for all of these distortions accurately. The primary reason for poor ASR performance is a mismatch between training and test conditions. As a result, many methods have been proposed in the literature to reduce this mismatch and improve performance. These methods can be grouped under two main categories: feature enhancement methods and model adaptation methods. Methods in the former category aim to clean the features observed at test time so that they better match the trained models, whereas methods latter category adapt the acoustic models to the noise conditions of the test utterance. Examples of the feature enhancement methods include spectral subtraction and cepstral mean normalization (CMN). Examples of model adaptation methods include parallel model combination (PMC), MAP adaptation, and vector Taylor series (VTS)-based model adaptation [1, 2].

While feature enhancement methods can improve recognition accuracy, they may also introduce undesirable residual error due to the inaccurate nature of the noise reduction algorithms which causes additional model mismatch. To tackle this problem, a training algorithm was proposed for multi-condition data where both the training and test data are processed in the same manner [3]. By processing the training data with the enhancement algorithm that will be applied at test time, the models can capture the effects of the residual error expected to be seen at test time. While this results in improvements over the conventional approach of simply compensating the test data to match a clean acoustic model, it has the drawback that it relies on point estimates of the enhanced features in both training and test, and no uncertainty about the denoising is reflected in the processing.

Model adaptation techniques have also been effective at improving recognition accuracy in noisy conditions and in fact, state-of-the-art performance on the Aurora 2 corpus was recently obtained with the VTS model adaptation proposed in [2]. One reason model domain can potentially be superior to feature domain methods is that they do not rely on point estimates of the features themselves, but rather compensate the probability distributions of the hidden Markov models (HMMs) directly. While the method in [2] has been shown to be highly effective, it has two main drawbacks: 1) it requires acoustic models trained from clean data, which makes it sub-optimal for tasks for which such data does not exist and 2) the adaptation algorithm still makes approximations that are not accounted for during training.

In this paper, we propose a new algorithm called noise adaptive training (NAT) to overcome the aforementioned weaknesses of the previous methods. It is motivated by the multi-condition training algorithm in [3] and the speaker adaptive training (SAT) algorithm in [4]. The NAT algorithm integrates environmental distortion normalization into the HMM training using a new formulation of the EM algorithm that incorporates the same VTS approximation used in the model adaptation technique in [2]. As an analogy, the proposed NAT algorithm has the same relationship to VTS model adaptation as SAT has to MLLR adaptation.

The rest of the paper is organized as follows. In Section 2, we review HMM adaptation using a VTS approximation. The proposed NAT algorithm for VTS model adaptation is detailed in Section 3. We present experimental results in Section 4 and finally, some concluding remarks in Section 5.

2. HMM ADAPTATION USING VTS

Let us assume that in the time domain the clean speech x[m] is corrupted by additive noise n[m] and channel distortion h[m]:

$$\text{y[m]=x[m]*h[m]+n[m]}, \tag{1}$$

and y[m] is the corrupted speech signal and $*$ represents convolution. In the MFCC domain, this is equivalent to [1]

$$y = x + h + C\log(1+\exp(C^{-1}(n-x-h))), \tag{2}$$

where C is the discrete cosine transform (DCT) matrix and C^{-1} is its pseudo-inverse, y, x, n, h are the MFCC vectors corresponding to distorted speech, clean speech, noise, and channel, respectively. The noise n has a Gaussian probability density function (PDF) with mean vector μ_n and covariance matrix Σ_n, and channel h has a PDF of the Kronecker delta function $\delta(h-\mu_h)$. The Jacobian of y in Eq. 2 with respect to x, h and n evaluated at a fixed point $(\mu_{x,0}, \mu_{h,0}, \mu_{n,0})$ can be expressed as follows:

$$\frac{\partial y}{\partial x}|_{(\mu_{x,0},\mu_{h,0},\mu_{n,0})} = \frac{\partial y}{\partial h}|_{(\mu_{x,0},\mu_{h,0},\mu_{n,0})} = G \tag{3}$$

$$\frac{\partial y}{\partial n}|_{(\mu_{x,0},\mu_{h,0},\mu_{n,0})} = F = I - G \tag{4}$$

*This work was done while the first author worked as an intern at Microsoft Research, Redmond.

where

$$G = C \cdot \mathrm{diag}\left(\frac{1}{1 + exp(C^{-1}(\mu_{n,0} - \mu_{x,0} - \mu_{h,0}))}\right) \cdot C^{-1} \quad (5)$$

and $\mathrm{diag}(.)$ in Eq. 5 represents the diagonal matrix whose elements equal to the value of the vector in the argument. Then, the nonlinear equation in Eq. 2 can be approximated by using a first order VTS expansion around the point $(\mu_{x,0}, \mu_{h,0}, \mu_{n,0})$ as follows:

$$\begin{aligned} y &\approx \mu_{x,0} + \mu_{h,0} + g_0 \\ &+ G(x - \mu_{x,0}) + G(h - \mu_{h,0}) + F(n - \mu_{n,0}), \end{aligned} \quad (6)$$

where

$$g_0 = C\log(1 + \exp(C^{-1}(\mu_{n,0} - \mu_{x,0} - \mu_{h,0}))) \quad (7)$$

By taking the expectation of Eq. 6, we can see that this expression is also valid in the model domain. Thus, we can write the mean vector $\mu_{y_{sm}}$ of the mth Gaussian of the sth HMM state as follows:

$$\begin{aligned} \mu_{y_{sm}} \approx \mu_{x_{sm},0} + \mu_{h,0} + g_{sm,0} + G_{sm}(\mu_{x_{sm}} - \mu_{x_{sm},0}) \\ + G_{sm}(\mu_h - \mu_{h,0}) + F_{sm}(\mu_n - \mu_{n,0}), \end{aligned} \quad (8)$$

Here, G, F and g_0 are functions of the mean of the mth Gaussian in the sth state of the generic HMM $\mu_{x_{sm},0}$. Assuming x and n are independent, and given the noise covariance Σ_n, the covariance matrix of the adapted HMM, $\Sigma_{y_{sm}}$, can be computed as follows:

$$\Sigma_{y_{sm}} \approx G_{sm}\Sigma_{x_{sm}}G_{sm}^T + F_{sm}\Sigma_n F_{sm}^T \quad (9)$$

Eq. 8 and 9 are applied only to the parameters that correspond to the static MFCC features. For the dynamic portions of the features, the following adaptation formulas have been used:

$$\mu_{\Delta y_{sm}} \approx G_{sm}\mu_{\Delta x_{sm}}, \quad (10)$$

$$\mu_{\Delta\Delta y_{sm}} \approx G_{sm}\mu_{\Delta\Delta x_{sm}}, \quad (11)$$

$$\Sigma_{\Delta y_{sm}} \approx G_{sm}\Sigma_{\Delta x_{sm}}G_{sm}^T + F_{sm}\Sigma_{\Delta n}F_{sm}^T \quad (12)$$

$$\Sigma_{\Delta\Delta y_{sm}} \approx G_{sm}\Sigma_{\Delta\Delta x_{sm}}G_{sm}^T + F_{sm}\Sigma_{\Delta\Delta n}F_{sm}^T \quad (13)$$

where the noise is assumed stationary so that $\mu_{\Delta n} = 0$, $\mu_{\Delta\Delta n} = 0$.

In the traditional VTS model adaptation, e.g. [1][2], it is assumed that the HMMs are trained from clean speech. For a given test utterance, maximum likelihood (ML) estimates of the noise and channel parameters are computed with the expectation-maximization (EM) algorithm using an iterative VTS approximation. These parameters are then used to adapt the clean speech model parameters using Eq. 8-13 and the utterance is redecoded with the adapted models. In this work, we use the VTS adaptation algorithm in [2] as the basis of our approach. This work improves on [1] in that the algorithm adapts both the means and variances of full feature vector, i.e. the static, delta, and delta-delta parameters.

3. NOISE ADAPTIVE TRAINING

Let us assume that there are I utterances in the multi-condition training set $\mathcal{Y} = \{Y^{(i)}\}_{i=1}^I$, and $Y^{(i)}$ is a sequence of T_i observations corresponding to ith utterance. In traditional ML HMM training, the parameters are estimated such that the resulting generic model Λ_Y maximizes the likelihood of the multi-condition training data.

In NAT, we assume that each utterance in the training set has an associated distortion model $\Phi^{(i)} = \{\mu_n^{(i)}, \Sigma_n^{(i)}, \mu_h^{(i)}\}$ that describes the additive noise and channel. The NAT algorithm seeks to find the distortion model parameters for all utterances $\Phi = \{\Phi^{(i)}\}_{i=1}^I$, and the underlying "pseudo-clean" model parameters Λ_X that jointly maximize the likelihood of the multi-condition data when the model Λ_X is transformed to the adapted HMM of $\Lambda_Y^{(i)}$. This can be written in the ML sense as:

$$(\Lambda_X, \Phi) = \underset{(\overline{\Lambda}_X, \overline{\Phi})}{\mathrm{argmax}} \prod_{i=1}^{I} \mathcal{L}(Y^{(i)}; \overline{\Lambda}_Y^{(i)}) \quad (14)$$

where $\overline{\Lambda}_Y^{(i)} = VTS(\overline{\Lambda}_X, \overline{\Phi}^{(i)})$ is the adapted HMM using the VTS as detailed in Section 2. In Eq 14, $(\overline{\Phi}, \overline{\Lambda}_X)$ and (Φ, Λ_X) are the old and new parameters set, respectively. The term "pseudo-clean" is used to indicate that the model defined by Λ_X is not necessarily equivalent to models trained with clean speech, but rather the model that maximizes the likelihood of the multi-condition training data when processed by the same VTS adaptation scheme that will be used at runtime.

In the NAT, we use a new EM algorithm that learns the distortion model parameters and the pseudo-clean speech model parameters iteratively. Thus, we start with the following EM auxiliary function:

$$Q(\Phi, \Lambda, \overline{\Phi}, \overline{\Lambda}) = \sum_{i=1}^{I} \sum_{t,s,m} \gamma_{tsm}^{(i)} \log(p(y_t^{(i)} | s_t = s, m_t = m, \Lambda, \Phi)) \quad (15)$$

where $\sum_{t,s,m}$ represents summation over frames, states, and Gaussians, and $\gamma_{tsm}^{(i)}$ is the posterior probability of the mth Gaussian in the sth state of the HMM for frame t of the ith utterance

$$\gamma_{tsm}^{(i)} = p(s_t = s, m_t = m | Y^{(i)}, \overline{\Lambda}, \overline{\Phi}). \quad (16)$$

In Eq. 15, $p(y_t^{(i)} | s_t = s, m_t = m, \Lambda, \Phi) \sim \mathcal{N}(y_t^{(i)}; \mu_{y_{sm}}^{(i)}, \Sigma_{y_{sm}}^{(i)})$. Note that $\mu_{y_{sm}}^{(i)}, \Sigma_{y_{sm}}^{(i)}$ are actually utterance-dependent since they are functions of distortion parameters for that utterance $\Phi^{(i)}$.

To update the mean values of the distortion parameters $\Phi^{(i)}$ in the M-step of the EM algorithm, we take the derivative of Q with respect to μ_n and μ_h and set the result to zero. Then, the update formulas in Eq. 17 and 18 are obtained. It is assumed that the noise is stationary, hence $\mu_{\Delta n} = 0$ and $\mu_{\Delta\Delta n} = 0$.

There is no closed form solution for the noise covariance matrices, so they are optimized iteratively using Newton's method according to the following update equation:

$$\Sigma_n^{(i)} = \Sigma_{n,0}^{(i)} - \left[\left(\frac{\partial^2 Q}{\partial^2 \Sigma_n^{(i)}}\right)^{-1}\left(\frac{\partial Q}{\partial \Sigma_n^{(i)}}\right)\right]_{\Sigma_n^{(i)} = \Sigma_{n,0}^{(i)}} \quad (22)$$

The noise covariance matrices for dynamic features $\Sigma_{\Delta n}^{(i)}, \Sigma_{\Delta\Delta n}^{(i)}$, are computed in a similar manner to Eq. 22 by replacing the static parameters with dynamic parameters. It is assumed that the noise covariance matrix is diagonal.

The pseudo-clean model parameters Λ_X are updated in a similar way to the distortion parameters except they are computed based on all utterances. Again, we take the derivative of Q with respect to $\mu_{x_{sm}}$ for static features (or $\mu_{\Delta x_{sm}}, \mu_{\Delta\Delta x_{sm}}$ for dynamic features) and set the result to zero. The update formulas in Eq. 19-21 are obtained to compute the model mean parameters. As with the noise covariance, since there is no closed form solution for computing the covariances of the HMM distributions, Newton's method is used to estimate them iteratively as follows:

$$\Sigma_{x_{sm}} = \Sigma_{x_{sm},0} - \left[\left(\frac{\partial^2 Q}{\partial^2 \Sigma_{x_{sm}}}\right)^{-1}\left(\frac{\partial Q}{\partial \Sigma_{x_{sm}}}\right)\right]_{\Sigma_{x_{sm}} = \Sigma_{x_{sm},0}} \quad (23)$$

The covariance matrices for dynamic features $\Sigma_{\Delta x_{sm}}, \Sigma_{\Delta\Delta x_{sm}}$ are computed in a similar way to Eq. 23 by replacing the static parameters with the dynamic parameters. Here, it is assumed that the covariance matrix is diagonal as in traditional acoustic model training.

The transition probabilities, the initial probabilities, and the mixture weights for the pseudo-clean model are computed in the same way as traditional ML training of the HMMs but using the new posterior probability as defined in Eq. 16. The NAT algorithm is summarized in the next section.

$$\mu_n^{(i)} = \mu_{n,0}^{(i)} + \left\{ \sum_{t,s,m} \gamma_{tsm}^{(i)} (F_{sm}^{(i)})^T (\Sigma_{y_{sm}}^{(i)})^{-1} F_{sm}^{(i)} \right\}^{-1} \cdot \left\{ \sum_{t,s,m} \gamma_{tsm}^{(i)} (F_{sm}^{(i)})^T (\Sigma_{y_{sm}}^{(i)})^{-1} (y_t^{(i)} - \mu_{x_{sm},0} - \mu_{h,0}^{(i)} - g_{sm,0}^{(i)}) \right\} \quad (17)$$

$$\mu_h^{(i)} = \mu_{h,0}^{(i)} + \left\{ \sum_{t,s,m} \gamma_{tsm}^{(i)} (G_{sm}^{(i)})^T (\Sigma_{y_{sm}}^{(i)})^{-1} G_{sm}^{(i)} \right\}^{-1} \cdot \left\{ \sum_{t,s,m} \gamma_{tsm}^{(i)} (G_{sm}^{(i)})^T (\Sigma_{y_{sm}}^{(i)})^{-1} (y_t^{(i)} - \mu_{x_{sm},0} - \mu_{h,0}^{(i)} - g_{sm,0}^{(i)}) \right\} \quad (18)$$

$$\mu_{x_{sm}} = \mu_{x_{sm},0} + \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T_i} \gamma_{tsm}^{(i)} (G_{sm}^{(i)})^T (\Sigma_{y_{sm}}^{(i)})^{-1} G_{sm}^{(i)} \right\}^{-1} \cdot \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T_i} \gamma_{tsm}^{(i)} (G_{sm}^{(i)})^T (\Sigma_{y_{sm}}^{(i)})^{-1} (y_t^{(i)} - \mu_{x_{sm},0} - \mu_{h,0}^{(i)} - g_{sm,0}^{(i)}) \right\} \quad (19)$$

$$\mu_{\Delta x_{sm}} = \mu_{\Delta x_{sm,0}} + \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T_i} \gamma_{tsm}^{(i)} (G_{sm}^{(i)})^T (\Sigma_{\Delta y_{sm}}^{(i)})^{-1} G_{sm}^{(i)} \right\}^{-1} \cdot \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T_i} \gamma_{tsm}^{(i)} (G_{sm}^{(i)})^T (\Sigma_{\Delta y_{sm}}^{(i)})^{-1} (\Delta y_t^{(i)} - G_{sm}^{(i)} \mu_{\Delta x_{sm,0}}) \right\} \quad (20)$$

$$\mu_{\Delta\Delta x_{sm}} = \mu_{\Delta\Delta x_{sm,0}} + \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T_i} \gamma_{tsm}^{(i)} (G_{sm}^{(i)})^T (\Sigma_{\Delta\Delta y_{sm}}^{(i)})^{-1} G_{sm}^{(i)} \right\}^{-1} \cdot \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T_i} \gamma_{tsm}^{(i)} (G_{sm}^{(i)})^T (\Sigma_{\Delta\Delta y_{sm}}^{(i)})^{-1} (\Delta\Delta y_t^{(i)} - G_{sm}^{(i)} \mu_{\Delta\Delta x_{sm,0}}) \right\} \quad (21)$$

3.1. NAT Algorithm

Step 1 Train the HMMs from multi-condition training data to initialize Λ_X. Initialize distortion parameters for each utterance such that the channel mean is set to zero and the noise mean and covariance are estimated from the first and last $N = 20$ frames (non-speech frames) of the utterance. Write distortion parameters, $\Phi^{(i)}$, for each utterance into a file.

Step 2 Read $\Phi^{(i)}$ from a file. Set VTS expansion point as $\mu_{x_{sm},0} = \mu_{x_{sm}}, \mu_{n,0} = \mu_n^{(i)}, \mu_{h,0} = \mu_h^{(i)}$. Adapt the HMM parameters with Eq. 8-13 to obtain $\Lambda_Y^{(i)}$.

Step 3 Compute the posterior probability in Eq.16.

Step 4 Update the distortion parameters $\Phi^{(i)}$ using Eq. 17-18 and 22, write them to a file.

Step 5 Accumulate the statistic for computing Λ_X: the matrix and the vector terms in Eq. 19-21 and 23 are updated for each utterance.

Step 6 If there are more utterances, go to **Step** 2, otherwise go to next step.

Step 7 Update Λ_X HMM parameters using Eq. 19-21 and 23.

These steps represent one iteration of the NAT algorithm. If the likelihood of all training data is increasing, we continue running additional iterations of training by going back to Step 2 until the likelihood converges. Once the pseudo-clean model parameters are learned, the distortion parameters Φ be discarded and the HMM parameters Λ_X are ready to be used with VTS adaptation at test time.

3.2. Discussion

In this section, we first compare the NAT with the SAT algorithm presented in [4]. The problem formulation of these two algorithms are quite similar with the following main difference: the SAT algorithm searches for a compact model Λ_c that will maximize the expected likelihood of the data from multiple speakers after performing MLLR transformation on Λ_c, whereas the NAT algorithm seeks for the pseudo-clean model Λ_x that will maximize the expected likelihood of the multi-condition data after adapted with the VTS algorithm. The variances are not updated in the SAT, hence we only focus on the comparison of the mean update equations here. The mean adaptation formula given in Eq. 8 can be written in the form of MLLR transformation as follows:

$$\mu_{y_{sm}}^{(i)} = W_{sm}^{(i)} * \mu_{x_{sm}} + \beta_{sm}^{(i)} \quad (24)$$

where $W_{sm}^{(i)} = G_{sm}^{(i)}$ and $\beta_{sm}^{(i)} = \mu_{x_{sm},0} + \mu_{h,0} + g_{sm,0} - G_{sm}\mu_{x_{sm},0}$, when the VTS expansion point is $\mu_{h,0} = \mu_h^{(i)}$ and $\mu_{n,0} = \mu_n^{(i)}$. Then, the model mean update equations for the SAT and NAT algorithms are in the same form of

$$\mu_{x_{sm}} = \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T_i} \gamma_{tsm}^{(i)} (W_{sm}^{(i)})^T (\Sigma_{y_{sm}}^{(i)})^{-1} W_{sm}^{(i)} \right\}^{-1} \cdot \left\{ \sum_{i=1}^{I} \sum_{t=1}^{T_i} \gamma_{tsm}^{(i)} (W_{sm}^{(i)})^T (\Sigma_{y_{sm}}^{(i)})^{-1} (y_t^{(i)} - \beta_{sm}^{(i)}) \right\} \quad (25)$$

with the following key exception: whereas SAT utilizes an unconstrained transformation matrix per speaker, NAT uses a matrix $W_{sm}^{(i)}$ that is specific to each Gaussian that is highly constrained by the utterance-specific distortion parameters.

Two other independent algorithms have been proposed previously with the motivation of training acoustic models using both clean and corrupted speech in [5] and [6]. In [5], a joint adaptive training (JAT) algorithm was proposed which was based on noise normalization using joint transforms for training models from noisy data. The formulation of the problem in JAT is different than ours, and does not take advantage of the well known nonlinear distortion model for the effect of additive noise and channel. In [6], a model training algorithm based on irrelevant variability normalization (IVN) was proposed. There are some important differences between IVN-based training and NAT. The proposed NAT algorithm seeks for Λ_X that matches best to the adaptation scheme performed at runtime, regardless of the true distribution of the clean speech, whereas IVN decouples Λ_Y, Λ_X and Φ while solving Eq 14 and tries to find the generic model Λ_X that best matches to the true distribution of the clean speech (irrespective of adaptation scheme performed at runtime). Also, the distortion and model parameters are solved in two separate steps in IVN, whereas all the parameters are solved within a single step in NAT.

4. EXPERIMENTS AND RESULTS

To verify the effectiveness of the proposed NAT method, a series of experiments were conducted on the Aurora 2 and Aurora 3 connected digit recognition corpora. The Aurora 2 consists of data degraded with additive noise and channel distortion [7]. Three test sets provided with the task are contaminated with noise types seen in the training data (Set A), unseen in the training data (Set B), and additive noise plus channel distortion (Set C). The acoustic models were trained using the standard "complex back end" Aurora 2 recipe. To examine the performance of our algorithm in real data, we also used the Aurora 3 for the experiments. The Aurora 3 consists of noisy digit recognition under realistic car environments [8], and contains three experiment conditions: well-matched (WM), medium-

Table 1. Word accuracy for each set of Aurora 2 using models trained on multi-condition data.

Method	Set A	Set B	Set C	Ave.
Baseline	91.68	89.74	88.91	90.35
CMN	92.97	92.62	93.32	92.90
CMVN	93.80	93.09	93.70	93.50
AFE	93.74	93.26	92.21	93.24
VTS	92.20	91.87	93.37	92.30
NAT	93.66	93.77	93.89	93.75

Table 2. Word accuracy for each set of Aurora 2 using models trained on clean data.

Method	Set A	Set B	Set C	Ave.
Baseline	60.43	55.85	69.01	60.31
CMN	68.65	73.71	69.69	70.88
CMVN	84.46	85.55	84.84	84.97
AFE	89.27	87.92	88.53	88.58
VTS	92.61	92.87	92.76	92.75
NAT	92.79	93.26	92.59	92.94

matched (MM), and highly-mismatched (HM). The acoustic models were trained using the standard "simple back end" scripts included with the Aurora 3. For both Aurora 2 and 3, 39-dimensional MFCC features consisting of 13 cepstral features plus delta and delta-delta features are used in the experiments. The cepstral coefficient of order zero (C0) is used instead of log energy. The cepstra are computed based on the spectral magnitudes.

We compared performance obtained by the proposed method (denoted as NAT), and that of standard VTS model adaptation (denoted as VTS). As mentioned earlier, the NAT and the VTS perform the identical adaptation at test time and only differ in how the HMM parameters are trained. The HMMs are trained using the standard ML training for the VTS results, and using the proposed NAT algorithm described in Section 3 for the NAT results. We also compared the results obtained by several well-known algorithms including cepstral mean normalization (CMN), cepstral mean and variance normalization (CMVN), and the ETSI advanced front-end (AFE) [9]. The AFE is a good representation of state of the art in the feature enhancement style of processing on these tasks.

In Table 1, we present word accuracy results for Aurora 2 using multi-condition training data. The baseline results were obtained with no compensation. The proposed NAT method achieves 93.75% average word recognition accuracy, and outperforms all other methods. NAT provides 11.97% relative improvement over CMN, 3.85% relative improvement over CMVN, 7.54% relative improvement over AFE, and 18.83% relative improvement over the VTS method.

We also applied NAT to the ML trained acoustic models using clean data to check whether the results could be improved. The set of results obtained using clean training data is presented in Table 2 for Aurora 2. NAT provides a small improvement over the VTS model adaptation (92.75% vs. 92.94%) showing that the clean models are not really clean and the distortion model still has approximations which we can model in NAT. Also, when the acoustic models are trained with clean data, both VTS and NAT, perform substantially better than the front-end feature-enhancement methods under the noisy test conditions, and NAT achieves the highest accuracy.

The baseline results for the Aurora 3 are presented in Table 3 together with the results of the CMN, the CMVN and the AFE. In Aurora 3, there is no clean data available for training. Hence, the acoustic models are generated using the standard training data provided with the database for each experimental conditions. The proposed NAT algorithm achieves 90.66% average word recognition accuracy, and outperforms all other methods. NAT provides 39.23% relative improvement over CMN, 12.63% relative improvement over CMVN, 3.61% relative improvement over AFE, and 32.02% relative improvement over the VTS model adaptation.

Table 3. Word accuracy for the Aurora 3 experimental conditions

Method	Well	Mid	High	Ave
Baseline	91.34	78.40	55.84	77.94
CMN	92.97	84.43	71.57	84.63
CMVN	94.22	87.92	83.40	89.31
AFE	95.3	86.79	87.25	90.31
VTS	91.33	80.25	86.57	86.26
NAT	94.44	87.55	88.98	90.66

5. CONCLUSION AND FUTURE WORK

In this paper, we propose a noise adaptive training algorithm for noise robust automatic speech recognition. The NAT algorithm can use both clean and corrupted speech, and this is especially beneficial when there is no clean data available for training. We compared the performance of the NAT with the state-of-the-art (to the best of our knowledge) model adaptation (VTS) [2] and front-end feature cleaning on training and testing (AFE) [8], and demonstrated that the NAT performs better than both methods in the Aurora 2 and 3 tasks.

The current algorithm is based on the cepstral domain expression between clean and noisy speech in Eq. 2. This formulation assumes that there is zero cross correlation between speech and noise. Other researchers have shown that this term can be non-zero, so in the future, we hope to improve our algorithm by incorporating it into the algorithm. We also plan to apply the NAT algorithm to a large vocabulary task to see if it has any effect on state tying.

6. ACKNOWLEDGEMENT

We would like to thank Dr. Jinyu Li and Dr. Jasha Droppo at Microsoft for valuable discussions.

7. REFERENCES

[1] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM Adaptation Using Vector Taylor Series for Noisy Speech Recognition," in *Proc. of ICSLP*, Beijing, China, 2000.

[2] J. Li, L. Deng, D. Yu, Y. Gong, and A. Acero, "High-performance hmm adaptation with joint compensation of additive and convolutive distortions via Vector Taylor Series," in *Proc. of ASRU*, Kyoto, Japan, 2007.

[3] L. Deng, A. Acero, M. Plumpe, and X. Huang, "Large-Vocabulary Speech Recognition under Adverse Acoustic Environments," in *Proc. of ICSLP*, Beijing, China, 2000.

[4] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. of ICSLP*, Philadelphia, PA, 1996.

[5] H. Liao and MJF Gales, "Adaptive training with joint uncertainty decoding for robust recognition of noisy data," in *Proc. of ICASSP*, Honolulu, Hawaii, 2007.

[6] Y. Hu and Q. Huo, "Irrelevant variability normalization based HMM training using VTS approximation of an explicit model of enviromental distortions," in *Proc. of Interspeech*, 2007.

[7] H.G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. of ISCA ITRW ASR*, Paris, France, September 2000.

[8] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, J. Allen, and S. Euler, "Speechdat-car: a large speech database for automotive environments," in *Proc. of LREC*, Athens, Greece, 2000.

[9] D. Macho, L. Mauuary, B. Noé, Y.M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a Noise-Robust DSR Front-End on Aurora Databases," in *Proc. of ICSLP*, Denver, Colorado, 2002.