IMPROVEMENTS TO VTS FEATURE ENHANCEMENT

Jinyu Li, Michael L. Seltzer, and Yifan Gong

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052, U.S.A.

ABSTRACT

By explicitly modelling the distortion of speech signals, model adaptation based on vector Taylor series (VTS) approaches have been shown to significantly improve the robustness of speech recognizers to environmental noise. However, the computational cost of VTS model adaptation (MVTS) methods hinders them from being widely used because they need to adapt all the HMM parameters for every utterance at runtime. In contrast, VTS feature enhancement (FVTS) methods have more computation advantages because they do not need multiple decoding passes and do not adapt all the HMM model parameters. In this paper, we propose two improvements to VTS feature enhancement: updating *all* of the environment distortion parameters and noise adaptive training of the front-end GMM. In addition, we investigate some other performance-related issues such as the selection of FVTS algorithms and the spectrum domain that MFCC is extracted from. As an important result of our investigation, we established the FVTS method can achieve comparable accuracy as the MVTS method with a smaller runtime cost. This makes FVTS method an ideal candidate for real world tasks.

Index Terms— *VTS, feature enhancement, model adaptation, robust ASR*

1. INTRODUCTION

Environment robustness in automatic speech recognition (ASR) remains a difficult problem despite many years of research. The difficulty arises due to many possible types of distortions, including additive and convolutive distortions, which are not easy to predict accurately when developing the recognizers. In recent years, a model-domain approach that jointly compensates for additive and convolutive distortions (e.g., [1][2][3][4][5]) has yielded promising results. The various methods proposed so far use a parsimonious nonlinear *physical* model to describe the environmental distortion and use the vector Taylor series (VTS) approximation technique to find closed-form hidden Markov model (HMM) adaptation and noise/channel parameter estimation formulas. As shown in [5], VTS model adaptation achieves much better accuracy than other model adaptation technologies.

Although VTS model adaptation can achieve high accuracy, the computational cost is very high as all the Gaussian parameters in the recognizer need to be updated every time the environmental parameters (noise and/or channel) change. This time-consuming requirement hinders VTS model adaptation from being widely used, especially in large vocabulary continuous speech recognition (LVCSR) where the number of model parameters is large.

VTS feature enhancement has been proposed as a lower-cost alternative to VTS model adaptation. For example, a number of techniques have been proposed that can be categorized as model-based feature enhancement schemes [6][7]. These methods use a small GMM or HMM in the front end and the same methodology used in VTS model adaptation to derive a minimum mean squared error estimate of the clean speech features given the noisy observations. In addition to the advantage of low runtime cost, VTS feature enhancement can be easily combined with other popular feature-based technologies, such as HLDA, fMPE, etc., which are challenging to VTS model adaptation.

Recently, two improvements to VTS model adaptation have been introduced. First, a maximum likelihood updating of *all* of the environmental distortion parameters was proposed [4][5]. Significant improvements were obtained by updating the static and dynamic means and variances of noise and channel parameters rather than just their static means as was typically done previously. Second, a noise-adaptive training method was proposed that enabled models suitable for VTS adaptation to be trained from noisy training data [8]. This is a significant improvement as it removes the requirement that the model be trained from clean speech and enables VTS adaptation to be used in situations where systems are trained from noisy data typically captured from real-world deployed applications.

In this paper, we examine how these improvements to VTS model adaptation can be incorporated into VTS feature enhancement and whether they provide similar gains in accuracy. In addition, we highlight other algorithmic considerations that impact the performance of VTS feature enhancement including the order of the Taylor series expansion and the use of features derived from the magnitude spectrum versus the power spectrum.

The paper is organized as follows. Section 2 presents the VTS model adaptation (MVTS) method. Section 3 presents VTS feature enhancement (FVTS). Advanced technologies for improving FVTS are presented in Section 4. In Section 5, a number of experiments are performed to evaluate the performance of techniques proposed in this paper. Finally, we summarize our study and draw conclusions in Section 6.

2. VTS MODEL ADAPTATION

The nonlinear distortion model of speech signal in cepstral domain is [1]:

$$y = x + h + Clog\left(1 + \exp\left(C^{-1}(n - x - h)\right)\right), \quad (1)$$

where x, n, h, and y are the clean speech, noise, channel, and distorted speech, respectively, in the cepstral domain. By taking the expectation on both sides of Eq. (1) and use vector Taylor series (VTS) expansion, the static mean of the distorted speech signal μ_y is

$$\begin{aligned}\mu_y &= \mu_x + \mu_h + g(\mu_x, \mu_h, \mu_n) \\ &\approx \mu_x + \mu_{h,0} + G\left(\mu_h - \mu_{h,0}\right) + (I - G)\left(\mu_n - \mu_{n,0}\right),\end{aligned} \quad (2)$$

where

$$g(\mu_x, \mu_h, \mu_n) = Clog\left(1 + \exp\left(C^{-1}(\mu_n - \mu_x - \mu_h)\right)\right), \quad (3)$$

$\mu_x, \mu_h,$ and μ_n are the static mean of clean speech, noise, channel, and C is the DCT matrix. By noting,

$$G = \frac{\partial \mu_y}{\partial \mu_x} = C diag\left(\frac{1}{1+\exp(C^{-1}(\mu_n - \mu_x - \mu_h)}\right) C^{-1}, \quad (4)$$

$$\frac{\partial \mu_y}{\partial \mu_x} = I - G, \quad (5)$$

we can derive the MVTS adaption formulations for the static HMM parameters for the *k*-th Gaussian in the *j*-th state as (following [5]):

$$\mu_{y,jk} = \mu_{x,jk} + \mu_h + g(\mu_{x,jk}, \mu_h, \mu_n), \quad (6)$$

$$\Sigma_{y,jk} \approx diag\left(G(j,k)\Sigma_{x,jk}G(j,k)^T + (I - G(j,k))\Sigma_n(I - G(j,k))^T\right), \quad (7)$$

The dynamic HMM parameters can be adapted using the continuous time approximation [5]. We have proposed re-estimation formulas for the static noise and channel mean, and the static and dynamic noise variances in [5].

The implementation steps of the MVTS HMM adaptation algorithm described so far in this section and used in our experiments are summarized in the following:

1. Read in a distorted speech utterance;
2. Set the channel mean vector to all zeros;
3. Initialize the noise mean vector and diagonal covariance matrix using the first and last *N* frames from the utterance;
4. Compute the Gaussian-dependent *G*(.) with Eq.(4), and adapt the HMM parameters;
5. Decode the utterance with the adapted HMM parameters;
6. Re-estimate the noise and channel distortions using the above-decoded transcription;
7. Adapt the HMM parameters again;
8. Use the final adapted HMM model obtained in step 7 to decode the distorted speech feature and get output transcription.

3. VTS FEATURE ENHANCEMENT

In this section, we summarize how to enhance distorted speech features using FVTS. In contrast to MVTS, we use a GMM to represent the underlying speech space. The GMM is trained using all the training data.

1. Read in a distorted speech utterance;
2. Set the channel mean vector to all zeros;
3. Initialize the noise mean vector and diagonal covariance matrix using the first and last *N* frames from the utterance;
4. Compute the Gaussian-dependent *G*(.) with Eq.(4), and adapt the GMM parameters (Note that there is no state in GMM, therefore the (j,k) element in MVTS should now be denoted as the (k) element in FVTS);
5. Re-estimate the noise and channel distortions;
6. Adapt the GMM parameters again;
7. Use the adapted GMM model to create an MMSE estimate of the clean speech given the observed noisy speech Eq. (11) or Eq. (14);
8. Use the HMM model to decode the cleaned speech feature obtained in step 7 and get output transcription.

There is no more HMM adaptation step in this FVTS algorithm. Given that the number of model parameters in a GMM usually is smaller than that in an HMM, FVTS has significantly lower runtime cost. In the following, we present two FVTS algorithms that can be used in step 7.

In general, we can use the minimum mean square error (MMSE) method to get the estimate of clean speech

$$\hat{x}_{MMSE} = E(x|y) = \int_X xp(x|y)dx. \quad (8)$$

Suppose the clean-trained GMM is denoted as

$$p(x|\Lambda) = \sum_{k=1}^{K} c_k N(x; \mu_{x,k}, \Sigma_{x,k}),$$

together with Eq. (1), we have

$$\begin{aligned}\hat{x}_{MMSE} &= y - h - \int_X Clog\left(1 + \exp\left(C^{-1}(n - x - h)\right)\right)p(x|y)dx \\ &= y - h - \\ &\int_X Clog\left(1 + \exp\left(C^{-1}(n - x - h)\right)\right)\sum_{k=1}^{K} p(k|y)\, p(x|y,k)dx \\ &= y - h - \sum_{k=1}^{K} p(k|y) \int_X Clog\left(1 + \exp\left(C^{-1}(n - x - h)\right)\right)p(x|y,k)dx.\end{aligned} \quad (9)$$

Here, the Gaussian occupancy probability is calculated as

$$p(k|y) = \frac{c_k N(y; \mu_{y,k}, \Sigma_{y,k})}{\sum_{k=1}^{K} c_k N(y; \mu_{y,k}, \Sigma_{y,k})} \quad (10)$$

$\mu_{y,k}$ *and* $\Sigma_{y,k}$ are the adapted distorted speech static mean and variance of the *k*th component of the GMM. If we use the 0th-order VTS approximation for the nonlinear term in Eq. (9), we can get the MMSE estimation of cleaned speech x as

$$\hat{x}_{MMSE} = y - h - \sum_{k=1}^{K} p(k|y) Clog\left(1 + exp\left(C^{-1}(\mu_n - \mu_{x,k} - \mu_h)\right)\right). \quad (11)$$

This formulation was first proposed in [1], and we denote it as FVTS-0.

In [6], another solution was proposed when expanding Eq. (1) with the 1st-order VTS. For the *k*th component of GMM, the joint distribution of x and y is modeled as

$$\begin{bmatrix} x \\ y \end{bmatrix}_k \sim N\left(\begin{bmatrix} \mu_{x,k} \\ \mu_{y,k} \end{bmatrix}, \begin{bmatrix} \Sigma_{x,k} & \Sigma_{xy,k} \\ \Sigma_{yx,k} & \Sigma_{y,k} \end{bmatrix}\right) \quad (12)$$

With some Bayesian formulation, we have

$$E(x|y,k) = \mu_{x|y,k} = \mu_{x,k} + \Sigma_{xy,k}\Sigma_{y,k}^{-1}(x - \mu_{y,k})$$

With the 1st-order VTS expansion of Eq. (1) and the property that speech and noise are independent, it is easy to get

$$\Sigma_{xy,k} = \Sigma_{x,k} G_k^T. \quad (13)$$

Then we can get the MMSE estimate of clean speech as

$$\hat{x}_{MMSE} = E(x|y) = \sum_{k=1}^{K} p(k|y)E(x|y,k) = \sum_{k=1}^{K} p(k|y)\left(\mu_{x,k} + \Sigma_{x,k}G_k^T(\Sigma_{y,k})^{-1}(y - \mu_{y,k})\right). \quad (14)$$

We denote the solution of Eq. (14) as FVTS-1.

The key of FVTS is to get a reliable estimation of noise and channel distortion parameters, and accurately calculate the Gaussian occupancy probability. In contrast to Eq. (10), which only uses static feature to calculate the Gaussian occupancy probability, the static and dynamic features are used to get a more reliable Gaussian occupancy probability [9]

$$p(k|y, \Delta y, \Delta\Delta y) = \frac{c_k N(y; \mu_{y,k}, \mu_{\Delta y,k}, \mu_{\Delta\Delta y,k}, \Sigma_{y,k}\ \Sigma_{\Delta y,k}\ \Sigma_{\Delta\Delta y,k})}{\sum_{k=1}^{K} c_k N(y; \mu_{y,k}, \mu_{\Delta y,k}, \mu_{\Delta\Delta y,k}, \Sigma_{y,k}\ \Sigma_{\Delta y,k}\ \Sigma_{\Delta\Delta y,k})}, \quad (15)$$

which is plugged into Eqs. (11) and (14).

Regarding the runtime cost, MVTS needs to adapt HMM parameters twice (in Step 4 and 7), while FVTS needs to adapt GMM parameter twice (in Step 4 and 6). Usually, the number of parameters in the GMM is much smaller than that in the HMM.

Furthermore, two rounds of decoding (in Step 5 and 8) are needed in MVTS while only one round decoding (in Step 8) is performed in FVTS. As consequence, FVTS has much lower computational cost than MVTS.

4. IMPROVEMENTS TO FVTS

In this section, we show how recent improvements in MVTS can be incorporated into FVTS.

4.1 Updating all distortion parameters

In [1] and [6], only static noise and channel mean vectors are re-estimated:

$$\mu_n = \mu_{n,0} + \left(\sum_t \sum_k \gamma_t(k)\left(I - G(k)\right)^T \Sigma_{y,k}^{-1}\left(I - G(k)\right)\right)^{-1}$$
$$\left(\sum_t \sum_k \gamma_t(k)\left(I - G(k)\right)^T \Sigma_{y,k}^{-1}\left(y_t - \mu_{x,k} - \mu_{h,0} - g(\mu_{x,k}, \mu_{h,0}, \mu_{n,0})\right)\right)$$
$$\mu_h = \mu_{h,0} + \left(\sum_t \sum_k \gamma_t(k) G(k)^T \Sigma_{y,k}^{-1} G(k)\right)^{-1}$$
$$\left(\sum_t \sum_k \gamma_t(k) G(k)^T \Sigma_{y,k}^{-1}\left(y_t - \mu_{x,k} - \mu_{h,0} - g(\mu_{x,k}, \mu_{h,0}, \mu_{n,0})\right)\right)$$

In contrast, we propose to update all the distortion parameters. Here, we re-estimate the static noise variance by using a second-order approach

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

where Q is the EM auxiliary function of the current utterance [5]. The dynamic noise variances $\Sigma_{\Delta n}$ and $\Sigma_{\Delta\Delta n}$ are updated in a similar way. Note that the dynamic means of the channel and noise are assumed to be zero. This follows from the assumption that the channel is deterministic and the noise is stochastic but stationary. Please refer [5] for the detailed formulation.

After updating both the static and dynamic model parameters with the online distortion re-estimation, we can have a more accurate estimation of the Gaussian posterior probabilities.

4.2 Noise adaptive training of GMM

In FVTS, it is assumed that a GMM trained from clean speech is available. However, in real world tasks, sometimes it is hard to get clean training data that is otherwise matched to the speech expected to be seen in deployment. Therefore, the underlying GMM is trained from observed noisy speech, i.e. multi-condition training data. In this case, the physical model in Eq. (1) is no longer valid and FVTS should not be directly applied in theory. Noise adaptive training (NAT) [8] was proposed as a solution to this problem. NAT estimates a pseudo-clean canonical speech model from noisy training data by incorporating VTS model adaptation into the model training procedure. As an analogy, speaker adaptive training (SAT) starts from a speaker-independent model and iteratively updates the of the speaker transforms and the HMM parameters to estimate a canonical model with less speaker variability. In much the same way, NAT starts with a multi-condition model and iteratively updates the distortion parameters and the HMM parameters to estimate a canonical model with less environmental variability. For example, given an estimate of the distortion parameters of each utterance in the training set, the updated pseudo-clean mean vector can be expressed as

$$\mu_{x,k} = \mu_{x,k,0} + \left(\sum_i \sum_t \gamma_t^i(k) G^i(k)^T (\Sigma_{y,k}^i)^{-1} G^i(k)\right)^{-1}$$
$$\left(\sum_i \sum_t \gamma_t^i(k) G^i(k)^T (\Sigma_{y,k}^i)^{-1}\left(y_t^i - \mu_{x,k,0} - \mu_{h,0}^i - g(\mu_{x,k,0}, \mu_{h,0}^i, \mu_{n,0}^i)\right)\right)$$

where, i is the utterance index, $\mu_{x,k,0}$ is current value for the static mean of the kth Gaussian component and γ_t^i is the posterior probability. The update expressions for the dynamic means and static and dynamic variances can be similarly derived. The detailed process and formulas are described in [8]. While NAT was originally proposed for HMM training, it can be easily used for training a GMM from multi-condition training data that is suitable for use with FVTS.

5. EXPERIMENTS

The VTS algorithms presented in this paper are first evaluated on the standard Aurora 2 task [10]. The clean training set is used to train the standard "complex backend" HMM model [10], which has 3628 Gaussians. We also train a GMM with 552 Gaussians for FVTS. The test material consists of three sets of distorted utterances. Set-A and set-B contain eight different types of additive noise while set-C contains two different types of noise and additional channel distortion. Following the standard evaluation of Aurora 2, we report average accuracy which is the average of accuracy of all three test sets.

The acoustic features are 13-dimensional MFCCs, appended by their first- and second-order time derivatives. The cepstral coefficient of order 0 is used instead of the log energy in the original script.

The VTS algorithms presented in this paper are then used to adapt the above MLE HMMs or to enhance the distorted features utterance-by-utterance for the entire test set (Sets-A, B, and C). We use the first and last N=20 frames from each utterance for initial estimation of the noise means and variances.

5.1 Impact of online distortion estimation

In most literature [1][6], only static distortion parameter is re-estimated. In contrast, we updated all the distortion parameters so that a more reliable Gaussian occupancy probability (Eq. (15)) can be obtained. We evaluated the impact of online distortion estimation in Table 1. The MFCCs are computed from the power spectrum. The baseline accuracy (Acc.) on Aurora2 is 61.51%. It is clear that updating all of the distortion parameters is significantly better than updating only mean noise and channel parameters for both FVTS-0 (Eq. (11)) and FVTS-1 (Eq. (14)).

Table 1: Impact of re-estimation of the distortion parameters on Aurora2. The baseline model is trained with clean data.

Acc.	FVTS-0	FVTS-1
update static noise and channel mean parameters only [1][6]	86.72	84.69
update all mean and variance distortion parameters	88.61	86.08

5.2 Impact of spectrum domain for MFCC extraction

Table 2 summarizes the recognition accuracy of the baseline and two different FVTS methods with the MFCC features extracted from power spectrum and magnitude spectrum. For both features, FVTS-0 is better than FVTS-1, especially when the features are extracted from power spectrum. Also, it is clear that FVTS works better on MFCCs extracted from the magnitude spectrum rather than from the power spectrum. This is consistent with what has been observed in MVTS [5].

Table 2: Comparison of different FVTS methods and MFCC derived from different spectrum on Aurora2. The baseline model is trained with clean speech.

Acc.	Baseline	FVTS-0	FVTS-1
Power spectrum	61.51	88.61	86.08
Magnitude spectrum	50.64	89.71	89.60

5.3 Comparison of Different FVTS methods

From Table 1 and 2, we can see that FVTS-0 outperforms FVTS-1 in all setups. In the remaining experiments, we will only discuss FVTS-0 with MFCC features extracted from magnitude spectrum, which is the best option in Table 2.

5.4 Working with noisy training data

In this section, we study the performance of FVTS in the absence of clean training data. Two tasks are used for evaluation. One is Aurora2 with multi-condition training data. The HMM (containing 3628 Gaussians) and GMM (containing 552 Gaussians) models are trained with the same method as with the clean training data.

The second task is Aurora3 [11], which consists of noisy digit recognition under realistic car environments and contains three testing conditions: well matched, medium matched, and highly mismatched. The MFCC features are extracted using the same process as in Aurora2. The HMM model is trained using the standard "simple backend" script. All the Gaussians from this HMM model are collapsed to form the GMM used for FVTS.

Table 3 compares FVTS-0 with NAT-FVTS-0, which uses NAT to generate a pseudo-clean model to enhance the training and testing features with FVTS-0. Although breaking the assumption in Eq. (1), the accuracy of FVTS-0 is still better than the baseline. However, on both Aurora2 and Aurora3 tasks, FVTS-0 is much worse than NAT+FVTS-0, which is consistent with the assumption in Eq. (1) and should be the right way to work with noisy training data. Comparing Tables 2 and 3, it is evident that using multi-condition training data and NAT results in higher accuracy for FVTS (92.92%) that the traditional approach where the GMM and HMM are trained from clean speech (89.71%).

Table 3: Comparison of FVTS-0 and NAT + FVTS-0 with multi-condition training data

Acc.	Aurora2	Aurora3
Baseline	83.17	77.94
FVTS-0	87.35	84.05
NAT + FVTS-0	92.92	89.11

5.5 Accuracy gap between FVTS and MVTS

In Table 4, we compare the accuracy between FVTS and MVTS. With much better accuracy achieved than literature (e.g., [6]), FVTS presented in this paper has a comparable accuracy as MVTS. The tradeoff between accuracy and computation cost will determine which technology is more suitable to be used in the real world deployment scenario.

Table 4: Comparison of FVTS and MVTS on Aurora 2 and Aurora 3 when multi-condition training data and NAT are used.

Acc.	Aurora 2	Aurora 3
NAT + FVTS-0	92.92	89.11
NAT + MVTS	93.75	90.66

6. CONCLUSIONS

In this paper, we gave a comprehensive study on issues related with the VTS feature enhancement (FVTS) technologies. To improve FVTS, we incorporated recent advancements developed in VTS model adaptation (MVTS). In contrast to previous works, we re-estimate both static and dynamic distortion parameters and get more reliable Gaussian occupancy probability estimates. This enabled our FVTS methods to obtain much higher accuracy than the previous works (e.g., [6]). We also showed that additional gains in FVTS can be obtained by using multi-condition training data in conjunction with noise adaptive training to obtain a pseudo-clean canonical GMM. We also showed that the FVTS method with Eq. (11) is more effective than the method with Eq. (14) in dealing with noise. It was demonstrated that MFCC extracted from magnitude spectrum gives higher accuracy for FVTS, which is consistent with our discovery in MVTS. Finally, we highlighted the remaining accuracy gap between FVTS and MVTS on Aurora2 and Aurora3 tasks.

Several issues should be addressed in the future. First, the experiments reported in this paper were limited to digit recognition tasks. The computational advantage of FVTS over MVTS is significant when the number of GMM parameters is much smaller than that of HMM parameters, which is true in the LVCSR scenario. We will work on noisy LVCSR tasks to verify the effectiveness of FVTS. Second, in current study, we only use the standard VTS technology to update GMMs while it has been shown that VTS with phase-sensitive distortion [5] and unscented transform [12] technologies can help to improve the modeling quality. We will apply these technologies to FVTS in the future.

7. REFERENCES

[1] P. Moreno, *Speech Recognition in Noisy Environments*. PhD. Thesis, Carnegie Mellon University, 1996.

[2] M. J. F. Gales, *Model-Based Techniques for Noise Robust Speech Recognition*, PhD. Thesis, Cambridge University, 1995.

[3] A. Acero, et al., "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proc. ICSLP*, Vol.3, pp. 869-872, 2000.

[4] H. Liao and M. J. F. Gales, "Joint uncertainty decoding for robust large vocabulary speech recognition, *Tech. Rep. CUED/TR552*, University of Cambridge, 2006.

[5] J. Li, et al., "A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions," *Computer Speech and Language*, vol. 23, pp. 389-405, 2009.

[6] V. Stouten, et al., "Robust speech recognition using model-based feature enhancement," in *Proc. European Conference on Speech Communication and Technology*, pp. 17-20, 2003.

[7] J. Droppo, L. Deng, and A. Acero, "A comparison of three non-linear observation models for noisy speech features," in *Proc. Eurospeech*, 2003.

[8] O. Kalinli, M. L. Seltzer, and A. Acero, "Noise adaptive training using a vector Taylor series approach for robust automatic speech recognition," in *Proc. ICASSP*, 2009.

[9] V. Stouten, *Robust Automatic Speech Recognition in Time-varying Environments*, Ph.D. Thesis, K. U. Leuven, Leuven 2006.

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

[11] A. Moreno, et al., "Speechdat-car: a large speech database for automotive environments," in *Proc. LREC*, 2000.

[12] J. Li, et al., "Unscented transform with online distortion estimation for HMM adaptation," in *Proc. Interspeech*, 2010.