SECOND ORDER VECTOR TAYLOR SERIES BASED ROBUST SPEECH RECOGNITION

Suliang Bu[1] *Yanmin Qian*[1] *Khe Chai Sim*[2] *Yongbin You*[1] *Kai Yu*[1]

[1] MOE-Microsoft Key Lab. for Intelligent Computing and Intelligent Systems
Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Department of Computer Science, National University of Singapore
09210240096@fudan.edu.cn, {yanminqian,youyongbin,kai.yu}@sjtu.edu.cn, simkc@comp.nus.edu.sg

ABSTRACT

Vector Taylor Series (VTS) model based compensation approach has been successfully applied to various robust speech recognition tasks. In this paper, a novel method to derive the formula to calculate the static and dynamic statistics based on *second-order* VTS (sVTS) is presented, which provides a new insight on the VTS approximation. Lengthy derivation could therefore be avoided when high order VTS is used and the proposed approach is more compact and easier to implement compared to previous high order VTS approaches. Experiments on Aurora 4 showed that the proposed sVTS based model compensation approach obtained 16.7% relative WER reduction over traditional first-order VTS (fVTS) approach.

Index Terms— robust speech recognition, model based compensation, Vector Taylor Series

1. INTRODUCTION

It is known that the performance of automatic speech recognition (ASR) system degrades greatly when additive noise presents and the system is trained only with clean speech. Considering 1) retraining a large vocabulary continuous ASR system is generally time-consuming and 2) lots of data, which should be recorded in a specific noisy condition, are often unavailable, adapting original models using a small sample of test speech is welcomed. Generally almost all methods proposed previously can be grouped into two categories: feature enhancement approach and model compensation approach.

Feature enhancement approach tries to remove the effect of the noise in the test utterance so that the processed data could better match the models trained using clean data. Usually feature enhancement has less computational cost compared to model compensation. However, it has the drawback that it relies on point estimates of the enhanced features [1]. In contrast, model compensation approach adapts the models by compensating the probability distribution of previously trained models. Among these proposed methods, first-order *vector Taylor series* approach has been widely adopted [2, 3, 4] because of its simple formula and effectiveness. However, relatively large residual errors would be caused by so simply truncated Taylor series approximation.

It is believed that higher order VTS could further reduce the mismatch between the adapted models and observation. Therefore, some efforts has been given on this direction. In [5], high order VTS is approximated by a linear function, which aims to minimize the mean squared error; in [6], second order VTS is used to calculate the static mean of the noisy speech. Extending work in [6], [7] further derived the formula for the dynamic mean. As for [8], it approximates the mismatch function with any order VTS using feature enhancement method, but recursion operation is needed when high order is used. All these works, however, expand the mismatch function around two variables, the noise and the clean speech, which would be complex to derive the formula. In contrast, based on feature enhancement way, [9] expands it in log-spectral domain with respect to $\mathbf{n}^{\mathbf{l}} - \mathbf{x}^{\mathbf{l}}$ instead of $(\mathbf{n}^{\mathbf{l}}, \mathbf{x}^{\mathbf{l}})$, where $\mathbf{n}^{\mathbf{l}}$ and $\mathbf{x}^{\mathbf{l}}$ are log-spectra of noise and clean speech. However, they make the inaccurate assumption that components in log-spectral features are uncorrelated. In this paper, we use a novel way to derive the formula in cepstral domain to compute dynamic statistics as well as statics based on second order VTS. As will be shown, the accurate dynamics statistics are the key for a better performance. Furthermore, the relation between fVTS and sVTS is also described.

The paper is organized as follows. In section 2, we describe the formulation to calculate static and dynamic statistics. The relation between fVTS and sVTS is shown in section 3. Experimental results on aurora 4 are reported and analyzed in section 4 and finally we conclude the work in section 5.

2. MODEL-BASED COMPENSATION USING SVTS

In this study, only additive noise is considered, and the channel distortion is ignored. And the formulation to calculate static and dynamic statistics will be described respectively.

2.1. Formula to Calculate Static Statistics

For static features, the nonlinear effect of additive noise in cepstral domain can be expressed as :

This work was performed while the first author was an intern at SJTU. It was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61222208 and JiangSu NSF project No. 201302060012.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

$$\mathbf{y_s} = \mathbf{x_s} + g(\mathbf{n_s} - \mathbf{x_s}) \tag{1}$$

where $\mathbf{y_s}$, $\mathbf{x_s}$ and $\mathbf{n_s}$ are static features corresponding to noisy speech, clean speech and additive noise respectively and the subscript "s" indicates static parameters. Here both $\mathbf{x_s}$ and $\mathbf{n_s}$ are assumed to have Gaussian distributions with mean $\boldsymbol{\mu_{xs}}$, $\boldsymbol{\mu_{ns}}$ and covariance $\mathbf{\Sigma_{xs}}$, $\mathbf{\Sigma_{ns}}$, respectively. In Eq. (1),

$$g(\mathbf{m}) = \mathbf{C}\ln\left(\mathbf{1} + e^{\mathbf{C^{-1}m}}\right) \tag{2}$$

where $\mathbf{C}$ is the discrete cosine transform (DCT) matrix.

Taking expectation of the noisy speech, we would have:

$$E\left[\mathbf{y_s}\right] = E\left[\mathbf{x_s}\right] + \mathbf{C}\cdot E\left[\ln\left(\mathbf{1} + \mathrm{e}^{\mathbf{C^{-1}}(\mathbf{n_s}-\mathbf{x_s})}\right)\right] \tag{3}$$

Usually, we will expand $g(\cdot)$ around $\mathbf{x_s}$ and $\mathbf{n_s}$. But it is less efficent to derive the formula especially high order VTS is used. Here we consider an alternative way. We use $\mathbf{z_s}$ to denote $\mathbf{C^{-1}}(\mathbf{n_s} - \mathbf{x_s})$. Since each component of $\mathbf{x_s}$ and $\mathbf{n_s}$ is Gaussian distributed and $\mathbf{C^{-1}}$ is a linear transformation, $\mathbf{z_s}$ is also Gaussian distributed. Please note that components in $\mathbf{z_s}$ are not mutually independent, nor $\mathbf{z_s}$ and $\mathbf{x_s}$ are independent. Comparing to Eq. (3), we now have:

$$E\left[\mathbf{y_s}\right] = E\left[\mathbf{x_s}\right] + \mathbf{C}\cdot E\left[\ln\left(\mathbf{1} + \mathrm{e}^{\mathbf{z_s}}\right)\right] \tag{4}$$

The second-order vector Taylor expansion of Eq. (1) is equivalent to the sum of each term's second-order Taylor expansion. For $\mathbf{x_s}$, its corresponding expansion is still $\mathbf{x_s}$. As for $g(\cdot)$, its corresponding expansion is much simplified because it now has only one random vector $\mathbf{z_s}$.

The speech and noise are assumed to be independent, then the mean and covariance for $\mathbf{z_s}$ are given by

$$\boldsymbol{\mu_{zs}} = \mathbf{C^{-1}}(\boldsymbol{\mu_{ns}} - \boldsymbol{\mu_{xs}}) \tag{5}$$

$$\mathbf{\Sigma_{zs}} = \mathbf{C^{-1}}(\mathbf{\Sigma_{ns}} + \mathbf{\Sigma_{xs}})(\mathbf{C^{-1}})^{\mathbf{T}} \tag{6}$$

Then the second-order vector Taylor series expansion of $\ln(\mathbf{1} + \mathrm{e}^{\mathbf{z_s}})$ around $\boldsymbol{\mu_{zs}}$ can be written as

$$\begin{aligned}\ln(\mathbf{1} + \mathrm{e}^{\mathbf{z_s}}) \approx\ & \mathbf{f}^{(0)} + \mathbf{f}^{(1)} \odot (\mathbf{z_s} - \boldsymbol{\mu}_{\mathrm{zs}}) \\ &+ \frac{1}{2}\mathbf{f}^{(2)} \odot (\mathbf{z_s} - \boldsymbol{\mu_{zs}}) \odot (\mathbf{z_s} - \boldsymbol{\mu_{zs}})\end{aligned} \tag{7}$$

where $\odot$ denotes element-by-element multiplication and

$$\mathbf{f}^{(0)} = \ln(\mathbf{1} + \mathrm{e}^{\boldsymbol{\mu_{zs}}}) \tag{8}$$

$$\mathbf{f}^{(1)} = \mathbf{1} - (\mathbf{1} + e^{\boldsymbol{\mu_{zs}}})^{-1} \tag{9}$$

$$\mathbf{f}^{(2)} = (\mathbf{1} + e^{\boldsymbol{\mu_{zs}}})^{-2} \odot e^{\boldsymbol{\mu_{zs}}} \tag{10}$$

where $\mathbf{f}^{(1)}$ and $\mathbf{f}^{(2)}$ correspond to the first and second derivatives of $\ln(\mathbf{1} + \mathrm{e}^{\mathbf{z_s}})$ at $\mathbf{z_s} = \boldsymbol{\mu_{zs}}$.

Computing the expectation of Eq. (7), we could have

$$E\left[\ln\left(\mathbf{1} + \mathbf{e}^{\mathbf{z_s}}\right)\right] \approx \mathbf{f}^{(0)} + \frac{1}{2}\mathbf{f}^{(2)} \odot diag^{-1}(\mathbf{\Sigma_{zs}}) \tag{11}$$

where $diag^{-1}(\cdot)$ denotes the operation of extracting the diagonal elements of a matrix as a column vector.

Together with Eq. (4) and (11), then the static mean of the noisy speech is given by

$$\boldsymbol{\mu_{ys}} \approx \boldsymbol{\mu_{xs}} + \mathbf{C}\cdot\left\{\mathbf{f}^{(0)} + \frac{1}{2}\left[\mathbf{f}^{(2)} \odot diag^{-1}(\mathbf{\Sigma_{zs}})\right]\right\} \tag{12}$$

As for the static covariance of noisy speech, it could be calculated by

$$\begin{aligned}\mathbf{\Sigma_{ys}} \approx E(\mathbf{y}_{\mathrm{s}} - \boldsymbol{\mu_{ys}})(\mathbf{y}_{\mathrm{s}} - \boldsymbol{\mu_{ys}})^T = \mathbf{\Sigma_{xs}} - \mathbf{K_1} \\ - \mathbf{K_1}^T + \mathbf{C}\cdot[\mathbf{\Sigma_{zs}} \odot \mathbf{F_1}]\cdot\mathbf{C}^T + \mathbf{K_2}\end{aligned} \tag{13}$$

where

$$\mathbf{K_1} = \mathbf{\Sigma_{xs}}(\mathbf{C}^{-1})^T \cdot diag(\mathbf{f}^{(1)}) \cdot \mathbf{C}^T \tag{14}$$

$$\mathbf{K_2} = \frac{1}{2}\mathbf{C}\cdot[\mathbf{\Sigma_{zs}} \odot \mathbf{\Sigma_{zs}} \odot \mathbf{F_2}]\cdot\mathbf{C}^T \tag{15}$$

$$\mathbf{F_1} = \mathbf{f}^{(1)}\left(\mathbf{f}^{(1)}\right)^T \tag{16}$$

$$\mathbf{F_2} = \mathbf{f}^{(2)}\left(\mathbf{f}^{(2)}\right)^T \tag{17}$$

In Eq. (14), $diag(\cdot)$ denotes the operation of generating diagonal matrix from a column vector.

2.2. Formula to Calculate Dynamic Statistics

To compute dynamic mean and covariance parameters, continuous time approximation [10] was used to derive the formula. In the following formula, we use the subscript "Δ" to denote delta statistics, and "$\Delta\Delta$" to denote delta delta.

Let's take the derivative of the approximation of $\mathbf{y_s}$ with respect to time, then we would have:

$$\frac{\partial\mathbf{y_s}}{\partial t} \approx \frac{\partial\mathbf{x_s}}{\partial t} + \mathbf{C}\cdot\left[\mathbf{f}^{(1)} \odot \frac{\partial\mathbf{z_s}}{\partial t} + \mathbf{f}^{(2)} \odot (\mathbf{z_s} - \boldsymbol{\mu_{zs}}) \odot \frac{\partial\mathbf{z_s}}{\partial t}\right] \tag{18}$$

Here we assume $\frac{\partial\mathbf{n_s}}{\partial t}$ and $\mathbf{n_s}$ are independent, similarly for speech. Since $\mathbf{C^{-1}}$ is a linear transformation, it is easy to prove $\frac{\partial\mathbf{z_s}}{\partial t}$ and $\mathbf{z_s}$ are independent, then we would have:

$$E\left[(\mathbf{z_s} - \boldsymbol{\mu_{zs}}) \odot \frac{\partial\mathbf{z_s}}{\partial t}\right] = E\left[\mathbf{z_s} - \boldsymbol{\mu_{zs}}\right] \odot E\left[\frac{\partial\mathbf{z_s}}{\partial t}\right] = 0 \tag{19}$$

Taking expectation of Eq. (18), we have

$$\boldsymbol{\mu_{y\Delta}} \approx E\left[\frac{\partial\mathbf{y_s}}{\partial t}\right] = \boldsymbol{\mu_{x\Delta}} + \mathbf{C}\cdot\left(\mathbf{f}^{(1)} \odot \boldsymbol{\mu_{z\Delta}}\right) \tag{20}$$

As for delta covariance, it could be computed by:

$$\begin{aligned}\mathbf{\Sigma_{y\Delta}} \approx \mathbf{\Sigma_{x\Delta}} - \mathbf{K_3} - \mathbf{K_3}^T + \mathbf{C}\cdot[\mathbf{\Sigma_{z\Delta}} \odot \mathbf{F_1}]\cdot\mathbf{C}^T \\ + \mathbf{C}\cdot\left[\mathbf{\Sigma_{zs}} \odot \left(\mathbf{\Sigma_{z\Delta}} + \boldsymbol{\mu_{z\Delta}}(\boldsymbol{\mu_{z\Delta}})^T\right) \odot \mathbf{F_2}\right]\cdot\mathbf{C}^T\end{aligned} \tag{21}$$

where

$$\mathbf{K_3} = \mathbf{\Sigma_{x\Delta}}(\mathbf{C^{-1}})^{\mathbf{T}} \cdot diag\left(\mathbf{f}^{(1)}\right) \cdot \mathbf{C}^T \tag{22}$$

Similarly, delta delta mean and covariance can be computed by:

$$\begin{aligned}\boldsymbol{\mu}_{\boldsymbol{y}\boldsymbol{\Delta}\boldsymbol{\Delta}} &\approx \boldsymbol{\mu}_{\boldsymbol{x}\boldsymbol{\Delta}\boldsymbol{\Delta}} + \mathbf{C}\cdot\left(\mathbf{f}^{(1)}\odot\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}\boldsymbol{\Delta}}\right)\\ &+\mathbf{C}\cdot\left[\mathbf{f}^{(2)}\odot\left(diag^{-1}(\boldsymbol{\Sigma}_{\mathbf{z}\boldsymbol{\Delta}})+\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}}\odot\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}}\right)\right]\end{aligned}\tag{23}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{y}\boldsymbol{\Delta}\boldsymbol{\Delta}} &\approx \boldsymbol{\Sigma}_{\mathbf{x}\boldsymbol{\Delta}\boldsymbol{\Delta}} - \mathbf{K_4} - \mathbf{K_4}^T\\ &+\mathbf{C}\cdot\left[\boldsymbol{\Sigma}_{\mathbf{z}\boldsymbol{\Delta}\boldsymbol{\Delta}}\odot\mathbf{F_1}+\mathbf{K_5}\odot\mathbf{F_2}\right]\cdot\mathbf{C}^T\end{aligned}\tag{24}$$

where

$$\mathbf{K_4} = \boldsymbol{\Sigma}_{\mathbf{x}\boldsymbol{\Delta}\boldsymbol{\Delta}}\left(\mathbf{C^{-1}}\right)^{\mathbf{T}}\cdot diag\left(\mathbf{f}^{(1)}\right)\cdot\mathbf{C}^T \tag{25}$$

$$\begin{aligned}\mathbf{K_5} &= \boldsymbol{\Sigma}_{\mathbf{zs}}\odot\left(\boldsymbol{\Sigma}_{\mathbf{z}\boldsymbol{\Delta}\boldsymbol{\Delta}}+\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}\boldsymbol{\Delta}}(\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}\boldsymbol{\Delta}})^T\right)\\ &+\boldsymbol{\Sigma}_{\mathbf{z}\boldsymbol{\Delta}}\odot\left(2\boldsymbol{\Sigma}_{\mathbf{z}\boldsymbol{\Delta}}+4\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}}(\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}})^T\right)\end{aligned}\tag{26}$$

In practice, covariance matrices are usually diagonalized for computational convenience, thus the decoder which has been optimized for diagonal covariance could be used.

In this paper, noise is modeled by a single Gaussian. To adapt clean trained models, noise parameters are needed. Usually, these parameters are estimated iteratively using EM-like algorithms. However, when high order VTS is used, it is not easy to derive the formula for re-estimation even for the mean. For comparison, in this study noise parameters are estimated by the first and last several frames of each test utterance.

3. RELATION BETWEEN FVTS AND SVTS

Since many people have implemented their first-order VTS with model compensation approach, it is interesting to investigate whether second-order approximation could be easily obtained by adding few terms on original formula. This section will describe the relation between them.

For static mean, the only difference between fVTS and sVTS is that the former only estimates the $\ln(\mathbf{1}+\mathrm{e}^{\mathbf{z_s}})$ by first-order approximation. According to [3], the static mean by first-order VTS is (ignoring channel distortion):

$$\boldsymbol{\mu}_{\boldsymbol{ys}} \approx \boldsymbol{\mu}_{\boldsymbol{xs}} + g(\boldsymbol{\mu}_{\boldsymbol{ns}} - \boldsymbol{\mu}_{\boldsymbol{xs}}) \tag{27}$$

The static mean by sVTS could be transformed as:

$$\boldsymbol{\mu}_{\boldsymbol{ys}} \approx \boldsymbol{\mu}_{\boldsymbol{xs}} + g(\boldsymbol{\mu}_{\boldsymbol{ns}} - \boldsymbol{\mu}_{\boldsymbol{xs}}) + \frac{1}{2}\mathbf{C}\cdot\left[\mathbf{f}^{(2)}\odot diag^{-1}(\boldsymbol{\Sigma}_{\mathbf{zs}})\right] \tag{28}$$

Comparing the above two formula, the second formula has an additional term in the end.

The static covariance by sVTS could be rewritten as:

$$\boldsymbol{\Sigma}_{\mathbf{ys}} \approx \mathbf{A}\boldsymbol{\Sigma}_{\mathbf{xs}}\mathbf{A}^T + (\mathbf{I}-\mathbf{A})\boldsymbol{\Sigma}_{\mathbf{ns}}(\mathbf{I}-\mathbf{A})^T + \mathbf{K_2} \tag{29}$$

where

$$\mathbf{A} = \mathbf{C}\cdot diag\left(\frac{1}{\mathbf{1}+exp\left(\mathbf{C}^{-1}(\boldsymbol{\mu}_{\boldsymbol{ns}}-\boldsymbol{\mu}_{\boldsymbol{xs}})\right)}\right)\cdot\mathbf{C}^{-1} \tag{30}$$

As for the delta features of fVTS, take the derivative of $\mathbf{y_s}$ with respect to time, it is

$$\frac{\partial\mathbf{y_s}}{\partial t} \approx \frac{\partial\mathbf{x_s}}{\partial t} + \mathbf{C}\cdot\left(\mathbf{f}^{(1)}\odot\frac{\partial\mathbf{z_s}}{\partial t}\right) \tag{31}$$

Thus, compared to Eq. (20), the delta mean of fVTS and sVTS are actually the same. The formula could be transformed as:

$$\boldsymbol{\mu}_{\boldsymbol{y}\boldsymbol{\Delta}} \approx \mathbf{A}\boldsymbol{\mu}_{\boldsymbol{x}\boldsymbol{\Delta}} + (\mathbf{I}-\mathbf{A})\boldsymbol{\mu}_{\boldsymbol{n}\boldsymbol{\Delta}} \tag{32}$$

As for the delta covariance and delta delta mean and covariance, they could be rewritten as the following:

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{y}\boldsymbol{\Delta}} &\approx \mathbf{A}\boldsymbol{\Sigma}_{\mathbf{x}\boldsymbol{\Delta}}\mathbf{A}^T + (\mathbf{I}-\mathbf{A})\boldsymbol{\Sigma}_{\mathbf{n}\boldsymbol{\Delta}}(\mathbf{I}-\mathbf{A})^T\\ &+\mathbf{C}\cdot\left[\boldsymbol{\Sigma}_{\mathbf{zs}}\odot\left(\boldsymbol{\Sigma}_{\mathbf{z}\boldsymbol{\Delta}}+\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}}(\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}})^T\right)\odot\mathbf{F_2}\right]\cdot\mathbf{C}^T\end{aligned}\tag{33}$$

$$\begin{aligned}\boldsymbol{\mu}_{\boldsymbol{y}\boldsymbol{\Delta}\boldsymbol{\Delta}} &\approx \mathbf{A}\boldsymbol{\mu}_{\boldsymbol{x}\boldsymbol{\Delta}\boldsymbol{\Delta}} + (\mathbf{I}-\mathbf{A})\boldsymbol{\mu}_{\boldsymbol{n}\boldsymbol{\Delta}\boldsymbol{\Delta}}\\ &+\mathbf{C}\cdot\left[\mathbf{f}^{(2)}\odot\left(diag^{-1}(\boldsymbol{\Sigma}_{\mathbf{z}\boldsymbol{\Delta}})+\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}}\odot\boldsymbol{\mu}_{\boldsymbol{z}\boldsymbol{\Delta}}\right)\right]\end{aligned}\tag{34}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{y}\boldsymbol{\Delta}\boldsymbol{\Delta}} &\approx \mathbf{A}\boldsymbol{\Sigma}_{\mathbf{x}\boldsymbol{\Delta}\boldsymbol{\Delta}}\mathbf{A}^T + (\mathbf{I}-\mathbf{A})\boldsymbol{\Sigma}_{\mathbf{n}\boldsymbol{\Delta}\boldsymbol{\Delta}}(\mathbf{I}-\mathbf{A})^T\\ &+\mathbf{C}\cdot\left[\mathbf{K_5}\odot\mathbf{F_2}\right]\cdot\mathbf{C}^T\end{aligned}\tag{35}$$

When noise is stationary, $\boldsymbol{\mu}_{\boldsymbol{n}\boldsymbol{\Delta}}$ and $\boldsymbol{\mu}_{\boldsymbol{n}\boldsymbol{\Delta}\boldsymbol{\Delta}}$ would be set to zero. In other words, the dynamic noise mean is not used to calculate the dynamic mean of noisy speech. Thus, comparing the formula in [3] with Eq. (28), (29), (32)-(35) and ignoring channel distortion, we can see that all statistics need only one respective term if sVTS is used instead of fVTS, except for delta mean, which keeps unchanged.

4. EXPERIMENTS AND RESULTS

4.1. Experimental Setup

In order to verify the effect of the proposed approach, we conducted experiments on aurora 4, which is based on the Wall Street Journal 5k database. In this study, speech models were trained on clean training data, which comprises 7138 training utterance. Decision tree state clustering was used to get about 3000 tied triphone states. Since this paper only considers additive noise, speech recognition experiments were conducted on test set B of aurora 4 corpus, which was recorded using the same microphone as the training data did. Therefore, channel distortion could be omitted. Six different noises at various SNRs were artificially added to turn original clean data into the noisy database. Each noise condition has 330 test utterances from 8 speakers. Only 16kHz testing data were used for evaluation. We used 12 Mel Frequency Cepstral Coefficient (MFCC) and C0 as well as the delta and delta-delta features. HTK [11] software was used to built the system, in which bigram language model was adopted. Each speech state was represented by 16 Gaussian components while 32 Gaussian components were used for the silence state model. Both main and relative beam pruning thresholds are set to be 230.0.

4.2. Experimental Results

In the following experiments, we first construct the GMM-HMM baseline, which results are illustrated in Table 1. As expected, the speech recognition system with clean trained models degrades greatly when testing environment is noisy.

clean	car	babble	rest.	street	airport	train	avg
6.8	37.0	55.1	55.0	64.7	48.8	63.9	54.1

Table 1. WER (%) of the baseline system on test B of AURORA 4 using HMMs trained on clean speech.

Then we implemented various VTS based methods. We first compare the following three approaches: 1) both static and dynamic statistics are calculated using fVTS; 2) static statistics are calculated by second order approximation and dynamic statistics calculated based on first-order, which is denoted as "VTS*" in Table 2 and Table 3; 3) both static and dynamic statistics are based on sVTS. Please note that since it is difficult to re-estimate the noise parameters for high order VTS as said in previous section, here we only adopt simple noise estimation by using the first and last several frames. It is unfair to compare sVTS with fVTS with noise re-estimation since simply one iteration of noise parameters could result in a large WER reduction. More accurate noise parameters estimation for sVTS will be in our future work.

	car	babble	rest.	street	airport	train	avg
fVTS	15.2	25.9	33.1	27.7	26.4	28.7	26.2
VTS*	14.8	25.1	31.7	26.9	25.5	28.1	25.3
sVTS	**11.6**	**21.3**	**26.1**	**22.2**	**20.1**	**21.8**	**20.5**

Table 2. WER (%) of several methods on test B when the first and last 10 frames is used to estimate noise.

	car	babble	rest.	street	airport	train	avg
fVTS	14.9	21.3	28.3	24.4	22.0	25.6	22.8
VTS*	14.7	21.1	27.2	23.8	21.7	24.9	22.2
VTS#	13.2	19.6	25.2	21.3	20.0	22.3	20.3
sVTS	**11.5**	**18.7**	**23.7**	**20.5**	**18.3**	**21.6**	**19.0**

Table 3. WER (%) of several methods on test B when the first and last 20 frames is used to estimate noise.

As Table 2 and 3 illustrated, we get large improvement by the VTS based approaches, and the WER of our fVTS system on test B in Table 3 (22.8%) is similar to the result in [12] (22.4%) when noise parameters are initially estimated by the first and last 20 frames.

Comparing fVTS and VTS* in Table 2 and 3, where both dynamic statistics are calculated based on the first order approximation, we find more accurate static statistics could improve the performance by a relatively small percentage, especially in Table 3. On the contrary, comparing VTS* and sVTS, it seems more accurate dynamic statistics are the key point to have a better performance, which is in accordance with the observation in [4].

From Table 2 and 3, sVTS achieves about 19.0% and 14.4% relative WER reduction over VTS* respectively. The comparison result indicates the dynamic statistics based on first-order approximation are not accurately estimated. The relatively large residual errors caused by such approximation might be the reason. The poor dynamic estimation also indicates the importance to search for a better way to calculate dynamic statistics. We find when all statistics are calculated using second-order approximation, the system obtains an even larger improvement when compared to fVTS: From table 2 and 3, the sVTS method gets about 21.8% and 16.7% relative WER reduction over fVTS respectively.

Using the method in this study, it is very easy to derive the formula for the mean, even for much high order. Therefore, it is necessary to investigate the performance when mean is calculated by higher order VTS and the covariance is based on lower order. In Table 3, "VTS#" denotes the way where mean is calculated by second-order approximation and covariance is first order based. Comparing fVTS and VTS#, we find the mere mean modification gets 2.5% absolute WER reduction on average. On the contrary, sVTS gets only 1.3% absolute reduction over VTS#. That is to say, the mean calculated by second-order approximation contributes more to the improvement than its corresponding covariance does, which suggests we could use more accurate mean to improve performance.

5. CONCLUSION

This paper presents the derivation of the second-order VTS (sVTS) using an alternative way. The method is rather general, therefore higher-order VTS could be easily derived. The main difference between this study and the work in [5, 6, 7, 8] is that we only need to expand the mismatch function around one random vector, so clear and compact formula could be got and potential computation cost would be reduced. As for [9], it assumes independence between different components in log-spectral features, which is a weakness. Instead, we derive the formula in cepstral domain. Besides, the formula to calculate the dynamics is also given, which is the key to improve the performance as is shown in experiment. Another difference is that the clear relation between sVTS and fVTS is provided, thus fVTS could be easily transformed into sVTS, which is especially useful for the people who has already implemented their first order VTS.

The effect of the proposed sVTS has been confirmed on the aurora 4 based robust speech recognition task. In our experiment sVTS got 21.8% and 16.7% relative WER reduction over fVTS when noise esimation is based on first and last 10 and 20 frames of each utterance, respectively. And during our experiment, we find the dynamic statistics are the key to have a better performance when VTS model compensation approach is used.

6. REFERENCES

[1] Ozlem Kalinli, Michael L Seltzer, and Alex Acero, "Noise adaptive training using a vector taylor series approach for noise robust automatic speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3825–3828.

[2] Pedro J Moreno, *Speech recognition in noisy environments*, Ph.D. thesis, Carnegie Mellon University, 1996.

[3] Alex Acero, Li Deng, Trausti T Kristjansson, and Jerry Zhang, "Hmm adaptation using vector taylor series for noisy speech recognition.," in *INTERSPEECH*, 2000, pp. 869–872.

[4] Jinyu Li, Li Deng, Dong Yu, Yifan Gong, and Alex Acero, "High-performance hmm adaptation with joint compensation of additive and convolutive distortions via vector taylor series," in *IEEE Workshop on Automatic Speech Recognition & Understanding*, 2007, pp. 65–70.

[5] Nam Soo Kim, "Statistical linear approximation for environment compensation," *IEEE Signal Processing Letters*, vol. 5, no. 1, pp. 8–10, 1998.

[6] Veronique Stouten, *Robust automatic speech recognition in time-varying environments*, Ph.D. thesis, Katholieke Universiteit Leuven, 2006.

[7] Haitian Xu and KK Chin, "Joint uncertainty decoding with the second order approximation for noise robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3841–3844.

[8] Jun Du and Qiang Huo, "A feature compensation approach using high-order vector taylor series approximation of an explicit distortion model for noisy speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2285–2293, 2011.

[9] Guo-Hong Ding and Bo Xu, "Exploring high-performance speech recognition in noisy environments using high-order taylor series expansion.," in *INTERSPEECH*, 2004.

[10] P. S. Gopalakrishnan S. Balakrishnan-Aiyer R. A. Gopinath, M. J. F. Gales and M. A. Picheny, "Robust speech recognition in noise - performance of the IBM continuous speech recognizer on the ARPA noise spoke task," in *ARPA Workshop on Spoken Language System Technology*, 1995, pp. 127–130.

[11] Steve Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Valtcho Valtchev, and Phil Woodland, "The htk book," *Cambridge University Engineering Department*, vol. 3, pp. 175, 2002.

[12] Liang Lu, KK Chin, Arnab Ghoshal, and Steve Renals, "Joint uncertainty decoding for noise robust subspace gaussian mixture models," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 6, pp. 1–29, 2012.