# Vector Taylor Series based Joint Uncertainty Decoding

*Haitian Xu*

Department of Communication Technology,
Aalborg University, Denmark
hx@kom.aau.dk

*Luca Rigazio and David Kryze*

Panasonic San Jose Laboratory
550 S. Winchester Blvd, San Jose
{rigazio, kryze}@research.panasonic.com

## Abstract

Joint uncertainty decoding has recently achieved promising results by using front-end uncertainty in the back-end in a mathematically consistent framework. One drawback of the method is that it relies on stereo-data or numerical algorithms, such as DPMC, which have high computational complexity and are difficult to deploy in real applications. We propose a Vector Taylor Series (VTS) approach to joint uncertainty decoding which provides a closed-form solution to the key problem of estimating the clean/noisy speech cross-covariance matrix. Our solution does not require stereo-data or numerical integration. We also propose a new strategy to deal with the cross-covariance matrix singularity. Experiments on Aurora2 show that VTS-based joint uncertainty decoding has similar accuracy compared to DPMC-based joint uncertainty decoding while being at least three times faster. Finally, VTS-based joint uncertainty decoding provided more than 2% absolute improvement when combined with our new strategy for cross-covariance singularity.

**Index Terms**: speech recognition, noise robustness, VTS, uncertainty decoding

## 1. Introduction

Noisy environments significantly degrade the performance of automatic speech recognition (ASR) systems, in particular when the acoustic models are trained with clean speech. The relatively low robustness against environmental noise has become a major obstacle for the widespread deployment of ASR technology.

One way to tackle this problem is to estimate the clean speech features with feature enhancement methods [1] [2], and use the enhanced features for recognition. Based on different estimation strategies, these methods only provide unbiased estimation of the clean features which carries estimation errors (uncertainty).

To improve this, recent efforts [3] [4] explore the effects of front-end uncertainty into the back-end. In [5], a mathematically consistent uncertainty decoding framework is introduced and applied with the front-end enhancement method SPLICE. Further refinements of this framework lead to joint uncertainty decoding (JUD) which show better recognition performance than SPLICE-based uncertainty decoding [6]. A key element of JUD is the cross-covariance matrix that models the relationship between clean and noisy speech. The estimation of this matrix requires either the stereo data [6] or data-driven parallel model combination (DPMC) [7]. In real applications, stereo data is not always available, and DPMC is computationally expensive.

In this paper, we uses a Vector Taylor Series (VTS) based approach to compute the cross-covariance matrix. Our method does not need stereo data and is computationally more efficient than the DPMC. We achieve this by first extending the VTS for the cepstral domain and the dynamic features. Then VTS approximation is applied to simplify the GMM adaptation and to compute the cross-covariance matrix. We show that the cross-covariance matrix computation with VTS has a similar form for static and dynamic features, leading to simple and computationally tractable form. On the Aurora 2 task [8], the proposed VTS-based JUD shows similar recognition accuracy to the DPMC-based JUD while being more than three times faster.

Furthermore, it is observed that the cross-covariance matrix can be singular when the Signal-to-Noise Ratio (SNR) is low. The improper inversion in JUD leads to a large number of insertions. [9] addresses this problem by limiting the minimal values in the cross-covariance matrix with a threshold. The threshold however needs to be experimentally decided. This paper presents a new mapping function which is free of parameter tuning. More than 2% absolute accuracy improvement is observed when applying the new mapping strategy in the experiments.

The remainder of this paper is as follows: section 2 gives an overview of the joint uncertainty decoding technique; section 3 introduces VTS-based JUD; in section 4, experimental results on Aurora 2 are presented and conclusions are finally drawn in section 5.

## 2. Overview of Joint Uncertainty Decoding

In the classical hidden Markov model (HMM) based ASR, the core part is the calculation of the HMM state emission probability modeled by the GMM:

$$p(x|S) = \sum_{m \in S} c_m N(x; \mu_m, \Sigma_m), \tag{1}$$

where $x$ is the clean speech feature, $S$ is the HMM state, and $N(x; \mu_m, \Sigma_m)$ is the Gaussian probability density function with mean $\mu_m$, covariance matrix $\Sigma_m$ and mixture weight $c_m$.

Given the noisy speech feature $y$, JUD directly computes the state emission probability of $y$:

$$p(y|S) = \int p(y|x)p(x|S)dx. \tag{2}$$

To solve the integration in Eq.(2), the conditional probability $p(y|x)$ needs to be properly modeled. By modeling the clean feature space with a front-end GMM, the relationship between clean and noisy speech is expressed in each mixture $s_i$ by assuming their joint distribution Gaussian:

$$p(y|x, s_i) = N(y; \mu_{y|x,s_i}, \Sigma_{y|x,s_i}) \quad (3)$$

and the probability in Eq.(2) can be written as:

$$p(y|S) = \int \sum_i P(s_i|y)p(y|x, s_i)p(x|S)dx \quad (4)$$

where $\mu_{y|x,s_i}$ and $\Sigma_{y|x,s_i}$ are respectively the conditional mean and variance matrix in the front-end mixture $s_i$. Eq.(4) can be expressed as [7]:

$$\begin{aligned} p(y|S) &= \sum_{m,i} c_m P(s_i|y) N(y; B_i\mu_m - B_i b_i, B_i\Sigma_m B_i^T + \\ &\qquad B_i\tilde{\Sigma}_i B_i^T) \\ &= \sum_{m,i} c_m P(s_i|y)|A_i| N(A_i y + b_i; \mu_m, \Sigma_m + \tilde{\Sigma}_i) \end{aligned} \quad (5)$$

and

$$\begin{aligned} A_i &= B_i^{-1} = \Sigma_x^i(\Sigma_{yx}^i)^{-1} \\ b_i &= \mu_x^i - A_i\mu_y^i \\ \tilde{\Sigma}_i &= A_i\Sigma_y^i {A_i}^T - \Sigma_x^i \end{aligned}$$

where $\mu_y^i$ and $\Sigma_y^i$ respectively denote the mean and variance of the noisy speech in the front-end GMM. These parameters can be obtained from the corresponding clean GMM parameters $\mu_x^i$ and $\Sigma_x^i$ by parallel model combination (PMC). To make the computation efficient, it was suggested in [6] to further simplify Eq.(5) by selecting the front-end mixture with the highest likelihood:

$$p(y|S) = \sum_m c_m N(A_{i^*} y + b_{i^*}; \mu_m, \Sigma_m + \tilde{\Sigma}_{i^*}) \quad (6)$$

where

$$i^* = arg \max_i P(s_i|y)$$

Notice from Eq.(6) that JUD estimates the clean speech features by $A_{i^*} y + b_{i^*}$ whereas the matrix $\tilde{\Sigma}_{i^*}$ represents the uncertainty of the estimation.

## 3. VTS based Joint Uncertainty Decoding

The most computationally expensive parts of JUD are the adaptation of the front-end GMM and the estimation of the cross-covariance matrix $\Sigma_{yx}^i$. The adaptation often involves PMC whereas the estimation requires stereo data or DPMC which are not realistic in real-life applications.

The VTS method proposed in [2] has shown a recognition performance similar to PMC while having a much lower computational complexity. The VTS-based GMM adaptation and cross-covariance matrix calculation are therefore expected to be a good substitute for PMC and DPMC. However, the original VTS works in the log-Mel domain with static features and needs to be extended to cepstral domain and dynamic features.

### 3.1. VTS for Cepstral and Dynamic Features

The effect of additive noise is non-linear in the cepstral domain. For static features, the relationship is:

$$y = x + g(x, n) = x + C\ln(1 + e^{C^{-1}(n-x)}) \quad (7)$$

where $C$ denotes the discrete cosine transformation matrix, $n$, $x$ and $y$ the static features for the noise, clean speech and noisy speech, respectively. In each front-end mixture $s_i$, the above non-linear relationship can be linearly approximated by the first-order Taylor series as:

$$\begin{aligned} y &\approx \mu_x^i + g(\mu_x^i, \mu_n) + W(x - \mu_x^i) \\ &\quad + (I - W)g(\mu_x^i, \mu_n)(n - \mu_n) \\ W &= I + \bigtriangledown_x g(\mu_x^i, \mu_n) \end{aligned} \quad (8)$$

where $\mu_n$ is the noise mean and $I$ the identity matrix.

It is reasonable to consider the delta and delta-delta features as the first and second derivative of the static features over time [10]. Thus, we can write for the dynamic features:

$$\begin{aligned} \triangle y \approx \triangle\mu_x + (W - I)g(\mu_x^i, \mu_n)\triangle\mu_x + (I - W)\triangle\mu_n \\ +W(\triangle x - \triangle\mu_x) + Z(x - \mu_x) \\ +(I - W)(\triangle n - \triangle\mu_n) - Z(n - \mu_n) \end{aligned} \quad (9)$$

$$\begin{aligned} \triangle\triangle y \approx W\triangle\triangle\mu_x + Z\triangle\mu_x + (I - W)\triangle\triangle\mu_n - Z\triangle\mu_n \\ +W(\triangle\triangle x - \triangle\triangle\mu_x) + 2Z(\triangle x - \triangle\mu_x) \\ +K(x - \mu_x) + (I - W)(\triangle\triangle n - \triangle\triangle\mu_n) \\ -2Z(\triangle n - \triangle\mu_n) - K(n - \mu_n) \end{aligned} \quad (10)$$

where

$$\begin{aligned} Z &= \frac{\partial W}{\partial t} = \frac{\partial W}{\partial \mu_x}\triangle\mu_x + \frac{\partial W}{\partial \mu_n}\triangle\mu_n \\ K &= \frac{\partial Z}{\partial t} \end{aligned}$$

### 3.2. Adaptation of the front-end GMM

Based on the VTS relationship derived, the mean values of the front-end GMM can be adapted as follows:

$$\mu_y^i \approx E(y|s_i) = \mu_x^i + g(\mu_x^i, \mu_n) \quad (11)$$

$$\triangle\mu_y^i \approx W\triangle\mu_x^i \quad (12)$$

$$\triangle\triangle\mu_y^i \approx W\triangle\triangle\mu_x^i + Z\triangle\mu_x \quad (13)$$

In this paper, we don't update the covariance matrix in the front-end GMM because the adaptation requires the estimation of the noise variance which can be highly unreliable [11]. We therefore assumes $\Sigma_y^i = \Sigma_x^i$.

### 3.3. Cross-covariance Matrix Estimation

Based on VTS in Eq.(8), the cross-covariance matrix $\Sigma^i_{yx}$ for static features is computed as follows:

$$\begin{aligned}\Sigma^i_{yx} &= E[(y-\mu^i_y)(x-\mu^i_x)^T|s_i] \\ &= E\{W(x-\mu^i_x)(x-\mu^i_x)^T \\ &+ (I-W)(N-\mu_n)(x-\mu^i_x)^T\} \\ &= W\Sigma^i_x \end{aligned} \quad (14)$$

Similarly, for dynamic features, the cross-covariance matrices are obtained from Eq.(9) and (10):

$$\triangle\Sigma^i_{yx} = W\triangle\Sigma^i_x \quad (15)$$

$$\triangle\triangle\Sigma^i_{yx} = W\triangle\triangle\Sigma^i_x \quad (16)$$

We believe using VTS is better than DPMC for two reasons. First, VTS gives a closed-form solution of the cross-covariance matrix and reduces the computational complexity compared to DPMC which relies on the simulation of hundreds of samples $x$ and $n$ for each GMM mixture.

Second, the cross-covariance matrices in Eq.(14)-(16) have similar forms for static, delta and delta-delta feature components. Thus, the $A_i$ matrix of Eq.(5) becomes:

$$A = \begin{pmatrix} W^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & W^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & W^{-1} \end{pmatrix} \quad (17)$$

Having the exactly same blocks for static, delta and delta-delta features in $A_i$ matrices makes it possible to reduce the complexity needed to compute the $A_i$ by up to one-third.

## 4. Experiments

Experiments are conducted on the Aurora 2 database [8] of connected digits. The database is divided into two training sets (clean and multi-condition) and three noisy testing sets. Test set A and B respectively include four types of additive noise with SNR ranging from 20 to 0 dB while set C also contains convolutional noise. In this paper, we use the clean training set to train the models and only test set A for the recognition test. Recognition is performed with HTK [12]. Each digit is modeled by 16 HMM states with three mixtures whereas the silence is modeled by 3 states each with 6 mixtures.

| SNR(dB) | Subway | Babble | Car | Exhibition | Average |
|---|---|---|---|---|---|
| 20 | 97.02 | 91.05 | 96.54 | 96.79 | 95.35 |
| 15 | 92.82 | 74.40 | 88.16 | 92.63 | 87.00 |
| 10 | 77.00 | 47.88 | 63.44 | 76.43 | 66.19 |
| 5 | 49.68 | 24.58 | 27.92 | 43.69 | 36.47 |
| 0 | 23.00 | 10.40 | 9.57 | 16.17 | 14.79 |
| **Average** | **67.90** | **49.66** | **57.13** | **65.14** | **59.96** |

Table 1: Recognition accuracy (%) for baseline

The front-end is a 13-dimensional MFCC with delta and delta-delta components. To facilitate PMC and DPMC implementation, the dynamic features are generated by simple difference [10] and the energy term used in MFCC is the 0th cepstral component, instead of the logarithm energy.

Compared to other JUD implementations [6] which use stereo data to estimate noise parameters $\mu_n$ and $\Sigma_n$, we adopt a simple noise estimation based on the first 10 non-speech frames of each utterance. The DPMC simulates 100 random feature vector pairs $(x, n)$ for each front-end GMM mixture, and the $\Sigma^i_{yx}$ and $A_i$ matrices are assumed diagonal.

Moreover, an important detail is how to deal with singularities of the cross-covariance matrix $\Sigma^i_{yx}$. It is noticed in [9] that the $\Sigma^i_{yx}$ can become singular at low SNRs, which may lead to a large number of insertions. To address this problem, [9] limits the minimal value of each diagonal component $\sigma^i_{yx,k}$ in the cross-covariance matrix $\Sigma^i_{yx}$ as:

$$\hat{\sigma}^i_{yx,k} = max(\sigma^i_{yx,k}, \rho_{th}\sigma^i_{y,k}\sigma^i_{x,k}) \quad (18)$$

where $\rho_{th}$ is a threshold that needs to be experimentally tuned. Alternatively, we propose the following mapping on each diagonal component $a_k$ that has the advantage of not being sensitive to parameter tuning:

$$\hat{a}_k = a_k \frac{2}{1+a_k} \quad (19)$$

Table 2 – 5 show that JUD with 128 front-end mixtures achieves significant improvement compared to the baseline in table 1. The JUD results are not as good as in [9] because the noise estimation in [9] is from stereo data and has no estimation errors.

DPMC and VTS have quite similar recognition accuracy for almost all the noise types and SNRs. Compared to the hard threshold in table 2 – 3 with the optimal $\rho_{th} = 0.9$, the mapping technique brings more than 2% improvement.

Fig.1 compares the complexity of VTS and DPMC-based JUD over varying number of mixtures. Notice that VTS-based JUD is more than three times faster than DPMC-based JUD.

| SNR(dB) | Subway | Babble | Car | Exhibition | Average |
|---|---|---|---|---|---|
| 20 | 97.54 | 97.94 | 98.24 | 97.96 | 97.92 |
| 15 | 95.52 | 95.95 | 96.33 | 95.59 | 95.85 |
| 10 | 90.94 | 90.87 | 88.61 | 89.60 | 90.01 |
| 5 | 77.40 | 71.37 | 66.69 | 74.02 | 72.37 |
| 0 | 46.79 | 39.78 | 30.63 | 44.12 | 40.33 |
| **Average** | **81.64** | **79.18** | **76.10** | **80.26** | **79.29** |

Table 2: Recognition accuracy (%) for DPMC-based JUD with thresholding $\rho_{th} = 0.9$

| SNR(dB) | Subway | Babble | Car | Exhibition | Average |
|---|---|---|---|---|---|
| 20 | 97.51 | 97.88 | 98.27 | 97.96 | 97.91 |
| 15 | 95.52 | 95.95 | 96.33 | 95.59 | 95.85 |
| 10 | 90.94 | 90.87 | 88.61 | 89.60 | 90.01 |
| 5 | 77.40 | 71.37 | 66.69 | 74.02 | 72.37 |
| 0 | 46.79 | 39.78 | 30.63 | 44.12 | 40.33 |
| **Average** | **81.63** | **79.17** | **76.11** | **80.26** | **79.29** |

Table 3: Recognition accuracy (%) for VTS-based JUD with thresholding $\rho_{th} = 0.9$

| SNR(dB) | Subway | Babble | Car | Exhibition | Average |
|---|---|---|---|---|---|
| 20 | 97.67 | 97.82 | 98.18 | 98.12 | 97.95 |
| 15 | 95.52 | 95.95 | 96.78 | 95.56 | 95.95 |
| 10 | 91.71 | 90.96 | 90.90 | 90.25 | 90.96 |
| 5 | 80.66 | 74.24 | 74.77 | 76.74 | 76.60 |
| 0 | 55.11 | 42.74 | 37.58 | 51.81 | 46.81 |
| **Average** | **84.13** | **80.34** | **79.64** | **82.50** | **81.65** |

Table 4: Recognition accuracy (%) for DPMC-based JUD with $A_n$ mapping

| SNR(dB) | Subway | Babble | Car | Exhibition | Average |
|---|---|---|---|---|---|
| 20 | 97.73 | 97.73 | 98.15 | 98.15 | 97.94 |
| 15 | 95.58 | 95.98 | 96.81 | 95.53 | 95.98 |
| 10 | 91.53 | 90.69 | 90.93 | 90.13 | 90.82 |
| 5 | 80.87 | 74.52 | 74.86 | 76.92 | 76.79 |
| 0 | 54.87 | 42.90 | 37.43 | 51.53 | 46.68 |
| **Average** | **84.12** | **80.36** | **79.64** | **82.45** | **81.64** |

Table 5: Recognition accuracy (%) for VTS-based JUD with $A_n$ mapping

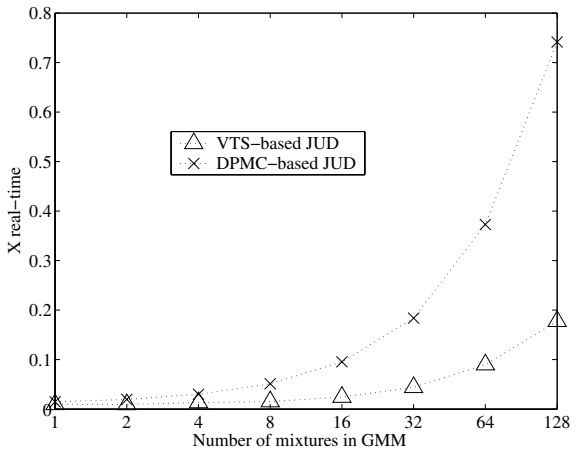


Figure 1: Real-time factor of VTS and DPMC-based JUD with different number of front-end GMM mixtures

## 5. Conclusions

Recently, joint uncertainty decoding, which estimates and propagates the front-end uncertainty to the back-end has achieved promising results. The algorithm however relies on the adaptation of the front-end GMM and the estimation of the cross-covariance matrix, which requires either stereo data or the complex numerical DPMC. In this paper, we overcome this drawback by introducing a VTS-based JUD and achieves closed-form solution for the cross-covariance matrix. This makes the computation of this matrix efficient. Also we introduce a new mapping strategy to tackle the sigularity problem of the cross-covariance matrix. On the Aurora 2 task, the proposed VTS-based method shows similar recognition performance while being more than three times faster compared to the DPMC-based JUD. In addition, more than 2% absolute improvement of the recognition accuracy can be achieved when combining the VTS-based method with the new mapping strategy.

## 6. Acknowledgment

The authors hereby appreciate the valuable discussions with H.Liao and M.J.F.Gales in Cambridge University.

## 7. References

[1] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 27, no. 2, pp. 112–120, April 1979.

[2] P.J.Moreno, *Speech Recognition in Noisy Environments*, Ph.D. thesis, CMU, 1996.

[3] N.B.Yoma and M.Villar, "Speaker verification in noise using a stochastic version of fthe weighted viterbi algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 158–166, March 2002.

[4] L.Deng, J.Droppo, and A.Acero, "Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 412 – 421, May 2005.

[5] J.Droppo, A.Acero, and L.Deng, "Uncertainty decoding with splice for noise robust speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Orlando, Florida, May 2002, pp. 57–60.

[6] H. Liao and M.J.F. Gales, "Joint uncertainty decoding for noise robust speech recognition," in *Proc. of INTERSPEECH 2005*, Lisbon, Portugal, Sep. 2005, pp. 3129–3132.

[7] H.Liao and M.J.F. Gales, "Uncertainty Decoding for Noise Robust Speech Recognition," Tech. Rep. CUED/F-INFENG/TR499, Cambridge University, Oct.2004.

[8] 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 2000*, Paris, France, Sep. 2000.

[9] H.Liao and M.J.F. Gales, "Issues with Uncertainty Decoding for Noise Robust Automatic Speech Recogntion," Tech. Rep. CUED/F-INFENG/TR499, Cambridge University, 2006.

[10] M.J.F.Gales, *Model-Based Techniques for Noise Robust Speech Recognition*, Ph.D. thesis, Cambridge University, 1995.

[11] J.C.Segura, A.de la Torre, M.C.Benitez, and A.M.Peinado, "Model-based compensation of the additive noise for continuous speech recognition: experiments using the Aurora-II database and tasks," in *Proc.of EuroSpeech*, Sep. 2001, pp. 221 – 224.

[12] S.Young, *HTK: Hidden Markov Model Toolkit V1.5*, 1993.