Efficient VTS Adaptation Using Jacobian Approximation

Jinyu Li, Michael L. Seltzer, and Yifan Gong

Microsoft Corporation, One Microsoft Way, Redmond, WA 98052

{jinyli; mseltzer; ygong}@microsoft.com

Abstract

By exploiting a model of environmental distortion, 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 more widely used. In this paper, we propose to reduce the computational cost of MVTS by replacing the Jacobian matrix used in the vector Taylor series approximation with a diagonal Jacobian matrix (DJVTS). We verify this approximation by showing that the Jacobian matrices are dominated by their diagonal elements and therefore the model distortion introduced by this approximation is very small. DJVTS gives similar accuracy as the standard MVTS method with significant reduction in computational cost. The proposed method also achieves higher accuracy than VTS-based feature enhancement.

Index Terms: vector Taylor series, Jacobian matrix, robust ASR

1. Introduction

It is well known that the accuracy of speech recognition systems degrades in noisy environments. This degradation is caused by a mismatch between the noisy speech seen in deployment and the speech used to train the recognizer. Acoustic model adaptation has been proposed as one way to reduce this mismatch and improve performance. The most successful methods of adaptation exploit a physical model of the relationship between clean speech and the observed speech. Because this relationship is nonlinear, model adaptation is performed using a vector Taylor series (VTS) approximation [1][2][3][4][5].

Although VTS model adaptation can achieve high accuracy, its computational cost is very high. In standard VTS model adaptation (MVTS), it is necessary to calculate a Jacobian matrix for each Gaussian distribution in the recognizer. These Gaussian-dependent Jacobians are necessary to adapt the HMM model parameters and to re-estimate the noise and channel distortion model parameters [5]. Computing the Jacobian matrix for all Gaussians requires a large number of matrix multiplications, which hinder MVTS from being more widely used.

As a solution, joint uncertainty decoding (JUD) was proposed to calculate the Jacobian matrices on a per-regression class basis instead of on a per-Gaussian basis [6]. Using fewer regression classes than Gaussians reduces the computational cost. VTS-JUD calculates a Jacobian matrix for each regression class but still adapts each Gaussian distribution individually [7]. Different from VTS-JUD, predictive constrained maximum likelihood linear regression (PCMLLR) is a method that applies the learned transforms to the feature space [7].

All of these methods save computation by estimating the Jacobian for each regression class rather than for each Gaussian. However, in VTS-JUD, there is no computation saving in the model adaptation step since the Jacobian matrices still need to be applied to all Gaussians. In PCMLLR, fewer Jacobian matrices are typically used but they need to be applied to the acoustic features of each frame.

VTS feature enhancement (FVTS) has been proposed as a lower-cost alternative to VTS model adaptation. A number of techniques categorized as model-based feature enhancement schemes have been proposed, e.g. [8][9]. These methods use a small GMM or HMM in the front end and the same methodology used in VTS adaptation to derive a minimum mean-squared error estimate of the clean speech features given the noisy observations. However, even after incorporating recent advances in VTS model adaptation into FVTS, the performance of FVTS still lags behind that of MVTS [10]. Thus, while FVTS has advantages in computational complexity, it comes at the price of reduced performance.

In this paper, we present a novel method to reduce the computational complexity of MVTS. Our approach specifically targets the Jacobian matrix as it represents the computational bottleneck in the VTS algorithm. In Section 2, we review VTS model adaptation in more detail and examine the computational complexity of the various steps in the algorithm. In Section 3, we perform an empirical analysis of the Jacobian matrices generated during adaptation. This analysis motivates an efficient approximation to the VTS algorithm that utilizes approximated Jacobian matrices with diagonal structure. The formulation of MVTS with diagonal Jacobian matrices (DJVTS) is given and the computational costs of standard MVTS and DJVTS are compared. In Section 4, we perform a series of experiments to show the efficacy of the proposed algorithm. Evaluated on Aurora2, the proposed DJVTS achieves accuracy comparable to standard MVTS with a significant reduction in computational cost. We also show that DJVTS also outperforms feature enhancement using FVTS. Finally, we summarize our study and conclude the paper in Section 5.

2. VTS Model Adaptation

In this section, we first briefly review the standard VTS model adaptation algorithm (MVTS) and then examine its computational cost.

2.1. Standard VTS Model Adaptation Algorithm

The nonlinear distortion model between clean and distorted speech can be expressed in the cepstral domain as [2]:

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

where x, n, h, and y are the clean speech, noise, channel, and distorted speech, respectively. C is the discrete cosine transform (DCT) matrix. By taking the expectation on both sides of (1), the static mean of the distorted speech signal μ_y can be written as

$$\mu_y = \mu_x + \mu_h + g(\mu_x, \mu_h, \mu_n) \quad (2)$$

where

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

and $\mu_x, \mu_n,$ and μ_h are the static cepstral means of clean speech, noise, channel, respectively. In VTS adaptation, the nonlinear function in (3) is approximated using vector Taylor series expansion. Using this approximation, the model parameters for Gaussian k in state j can be updated as (following [4]):

$$\mu_y(j,k) = \mu_x(j,k) + \mu_h + g(\mu_x(j,k), \mu_h, \mu_n), \quad (4)$$

$$\Sigma_y(j,k) \approx diag\left(G(j,k)\Sigma_x(j,k)G(j,k)^T + \left(I - G(j,k)\right)\Sigma_n\left(I - G(j,k)\right)^T\right), \quad (5)$$

where $G(j,k)$ is the Gaussian-dependent Jacobian matrix defined as

$$G(j,k) = \partial y/\partial x|_{\mu_x(j,k),\mu_n,\mu_h} = Cdiag\left(\frac{1}{1 + \exp\left(C^{-1}(\mu_n - \mu_x(j,k) - \mu_h)\right)}\right)C^{-1} \quad (6)$$

The delta and delta-delta parameters can be similarly updated (following [4]):

$$\mu_{\Delta y}(j,k) \approx G(j,k)\mu_{\Delta x}(j,k), \quad (7)$$

$$\mu_{\Delta\Delta y}(j,k) \approx G(j,k)\mu_{\Delta\Delta x}(j,k), \quad (8)$$

$$\Sigma_{\Delta y}(j,k) \approx diag\left(G(j,k)\Sigma_{\Delta x}(j,k)G(j,k)^T + \left(I - G(j,k)\right)\Sigma_{\Delta n}\left(I - G(j,k)\right)^T\right), \quad (9)$$

$$\Sigma_{\Delta\Delta y}(j,k) \approx diag\left(G(j,k)\Sigma_{\Delta\Delta x}(j,k)G(j,k)^T + \left(I - G(j,k)\right)\Sigma_{\Delta\Delta n}\left(I - G(j,k)\right)^T\right). \quad (10)$$

Significant improvements in the performance of MVTS can be obtained by re-estimating the distortion model parameters based on the first-pass decoding result. For example, the mean of the noise μ_n can be updated according to

$$\mu_n = \mu_{n,0} + A^{-1}b \quad (11)$$

$$A = \sum_{t,j,k}\gamma_t(j,k)\left(I - G(j,k)\right)^T\Sigma_y^{-1}(j,k)\left(I - G(j,k)\right)$$

$$b = \sum_{t,j,k}\gamma_t(j,k)\left(I - G(j,k)\right)^T\Sigma_y^{-1}(j,k)\left(y_t - \mu_x(j,k) - \mu_{h,0} - g\left(\mu_x(j,k), \mu_{h,0}, \mu_{n,0}\right)\right)$$

where $\gamma_t(j,k)$ is the posterior probability for Gaussian k in state j. The channel mean μ_h can be estimated similarly. To estimate the D-dimensional static noise variance vector $\Sigma_n = diag(\boldsymbol{\sigma}_n^2)$ with $\boldsymbol{\sigma}_n^2 = \left[\sigma_{n,1}^2, \sigma_{n,2}^2, \ldots \sigma_{n,D}^2\right]^T$, we use Newton's method, an iterative second order approach. Full derivations of the update formulae for the distortion model parameters can be found in [5].

2.2. Computational analysis of VTS adaptation

The computational cost of the steps in standard MVTS is analyzed in the following. Every element in the full $D * D$ dimension Jacobian matrix in (6) requires $\mathcal{O}(D)$ calculations, where D is the dimension of static cepstra. Therefore, for a system with a total of N Gaussians, the cost to calculate all Jacobian matrices is $\mathcal{O}(ND^3)$. When adapting the static mean with (4), every element in the mean vector requires $\mathcal{O}(D)$ calculations. Therefore, the total cost is $\mathcal{O}(ND^2)$. This is also true for the adaptation of other model parameters. In (11), $\left(I - G(j,k)\right)^T\Sigma_y^{-1}(j,k)\left(I - G(j,k)\right)$ is a full matrix and requires $\mathcal{O}(D^3)$ caluclations. Hence, the total cost is $\mathcal{O}(ND^3)$

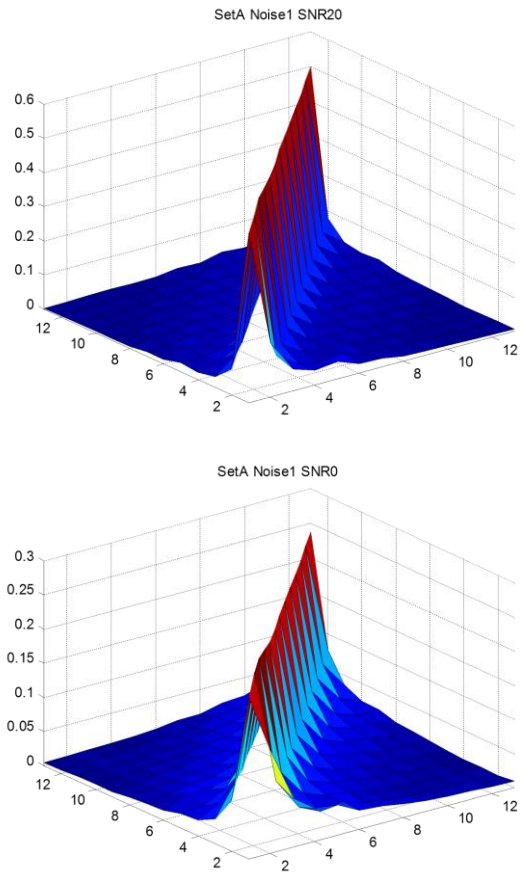

Figure 1: *The element-wise average of the absolute values of the Jacobian matrices in SNR20 and SNR0 conditions in Aurora 2 test set A with noise type 1 (subway noise).*

for re-estimation of the noise mean and channel mean. Re-estimation of the noise covariance also requires $\mathcal{O}(ND^3)$ calculations since the Hessian matrix has D^2 elements and each element needs $\mathcal{O}(ND)$ calculations [5].

3. VTS with Diagonal Jacobian Matrix

It is well-known that the DCT matrix has the orthogonality property, i.e. $C^T = C^{-1}$. With this property, the Jacobian $G(j,k)$ in (6) can be rewritten as

$$G(j,k) = Cdiag\left(\frac{1}{1 + \exp\left(C^{-1}(\mu_n - \mu_x(j,k) - \mu_h)\right)}\right)C^T \quad (12)$$

Since DCT matrix is used for decorrelation, an intuition is to approximate the off-diagonal components in (12) with 0 if we assume $diag\left(\frac{1}{1+exp\left(C^{-1}(\mu_n - \mu_x(j,k) - \mu_h)\right)}\right)$ is the covariance of some variables.

We verified this by analyzing the element-wise average of the absolute values of the Jacobian matrices obtained from the standard Aurora 2 task [11]. In Figure 1, we plot those values for two SNR conditions in test set A with noise type 1 (subway noise). In both conditions, we can see that the diagonal elements have much higher values than the off-diagonal elements, which suggests that a diagonal approximation of the Jacobian matrices is reasonable. Note that in the 0 dB SNR condition, the diagonal components are

not as dominant as in the 20 dB SNR condition. This means that the diagonal approximation is more accurate in high SNR conditions. Similar observations were also made for different types of noise.

Based on these observations, $G(j,k)$ can be approximated as a diagonal matrix:

$$G(j,k) = diag([G_{11}(j,k), G_{22}(j,k), \dots \dots G_{DD}(j,k)]) \quad (13)$$

Using (13), almost all matrix and vector operations in the original MVTS formulations (except (4)) can be reduced to scalar operations. This greatly reduces the computational cost. We refer to MVTS using the diagonal Jacobian approximation as DJVTS.

Using DJVTS, (5), (7)-(10) can be simplified to the following scalar operations with $d = [1, D]$

$$\sigma^2_{y,d}(j,k) = G^2_{dd}(j,k)\sigma^2_{x,d}(j,k) + \left(1.0 - G^2_{dd}(j,k)\right)^2 \sigma^2_{n,d} \quad (14)$$

$$\mu_{\Delta y,d}(j,k) = G_{dd}(j,k)\mu_{\Delta x,d}(j,k) \quad (15)$$

$$\mu_{\Delta\Delta y,d}(j,k) = G_{dd}(j,k)\mu_{\Delta\Delta x,d}(j,k) \quad (16)$$

$$\sigma^2_{\Delta y,d}(j,k) = G^2_{dd}(j,k)\sigma^2_{\Delta x,d}(j,k) + \left(1.0 - G_{dd}(j,k)\right)^2 \sigma^2_{\Delta n,d} \quad (17)$$

$$\sigma^2_{\Delta\Delta y,d}(j,k) = G^2_{dd}(j,k)\sigma^2_{\Delta\Delta x,d}(j,k) + \left(1.0 - G_{dd}(j,k)\right)^2 \sigma^2_{\Delta\Delta n,d} \quad (18)$$

The re-estimation of the distortion parameters can be similarly simplified. For example, the noise mean μ_n can be updated under DJVTS as

$$\mu_{n,d} = \mu_{n,d,0} + a_d / b_d \quad (19)$$

$$a_d = \sum_{t,j,k} \gamma_t(j,k)\left(1.0 - G_{dd}(j,k)\right)\left(y_{t,d} - \mu_{x,d}(j,k) - \mu_{h,d,0} - g_d\left(\mu_x(j,k), \mu_{h,0}, \mu_{n,0}\right)\right) / \sigma^2_{y,d}(j,k)$$

$$b_d = \sum_{t,j,k} \gamma_t(j,k)\left(1.0 - G_{dd}(j,k)\right)^2 / \sigma^2_{y,d}(j,k)$$

Updating the noise covariance is also simplified as it now only requires a vector of second-derivatives rather than the computation of the full Hessian matrix.

The comparison of the computational cost of the various steps in MVTS and DJVTS is shown in Table 1. With (13), only D elements need $\mathcal{O}(D)$ calculations. Therefore, for the a total of N Gaussians, the cost to calculate the Jacobian matrices is $\mathcal{O}(ND^2)$. Using a diagonal Jacobian approximation has no impact on static mean with (4). Therefore, the total cost to adapt static mean remains $\mathcal{O}(ND^2)$. However, the adaptation of other model parameters only requires $\mathcal{O}(ND)$ since these can now be done dimension by dimension. The re-estimation of the noise and channel mean can also be done with scalar manipulation as shown in (19). Hence, the cost is $\mathcal{O}(ND)$, with a relative reduction of $\mathcal{O}(D^2)$ from standard MVTS. With the noise variance update, only D elements are necessary to compute and every element only needs $\mathcal{O}(N)$ calculations. This can be seen by simplifying the noise covariance update expression in [5] with the diagonal Jacobian approximation. Therefore, the cost for updating the noise variance is also $\mathcal{O}(ND)$ in DJVTS.

Table 1: *Comparison of approximated computation cost of standard MVTS and DJVTS. N is the number of Gaussians, D is the dimension of the static cepstral feature vector.*

Processing Steps	**Standard MVTS**	**DJVTS**	**Relative Cost Reduction**
Calculate Jacobian matrix	$\mathcal{O}(ND^3)$	$\mathcal{O}(ND^2)$	$\mathcal{O}(D)$
Adapt static means	$\mathcal{O}(ND^2)$	$\mathcal{O}(ND^2)$	0
Adapt remaining model parameters	$\mathcal{O}(ND^2)$	$\mathcal{O}(ND)$	$\mathcal{O}(D)$
Estimate distortion model parameters	$\mathcal{O}(ND^3)$	$\mathcal{O}(ND)$	$\mathcal{O}(D^2)$

4. Experimental Evaluation

The proposed DJVTS algorithm presented in Section 3 has been evaluated on the Aurora 2 task [11] of recognizing digit strings subject to noise and channel distortions. The clean training set was used to train the HMMs using maximum likelihood estimation (MLE). The test material consists of three sets of distorted utterances. Sets A and B contain eight different types of additive noise while Set C contains two different types of noise plus additional channel distortion. Each type of noise is added into a subset of clean speech utterances, at seven different SNRs. This generates seven subgroups of test sets for a specified noise type, with clean, 20db, 15db, 10db, 5db, 0db, and -5db SNRs. The experimental setup follows the standard recipe provided by ETSI, using the standard complex backend [12] which generates HMMs with a total of 3628 Gaussians.

The features are 13-dimension MFCCs extracted from magnitude spectrum, appended by their first- and second-order time derivatives. The cepstral coefficient of order zero is used instead of the log energy.

The standard MVTS and DJVTS algorithm presented in this paper are used to adapt the MLE-trained HMMs utterance-by-utterance for the entire test set (Sets A, B, and C). The implementation steps described in [4] are used for all experiments. We use the first and last 20 frames from each utterance for initializing the noise means and variances and the channel mean is initialized to zero. For each utterance, a single iteration of distortion parameter re-estimation is performed.

We first performed an experiment to measure the accuracy of the diagonal Jacobian approximation. We evaluated the average KL divergence of the Gaussians adapted with DJVTS from the Gaussians adapted with standard VTS. The smaller this measure, the more accurate the approximation is. In addition to evaluating a diagonal Jacobian matrix, we also examined a banded matrix structure which includes the main diagonal and a number of additional diagonals on either side. The results are shown in Figure 2 for test data from 20 dB and 0 dB SNR with noise type 1 (subway noise). The x-axis shows the number of diagonals in the Jacobian matrix (the band-width of the matrix). The values at $x = 0$ represent the KL distance between the uncompensated clean model and standard VTS-adapted model using full Jacobians. The values at $x = 1$ show how far the models adapted with DJVTS are from those adapted with standard MVTS. As the width of the band increases (moving to the right on the x-axis in the figure), more diagonal bands in the Jacobian matrices are included. When $x = 13$, none of the elements is set as zero and the full Jacobian is used. This is equivalent to MVTS and therefore, the KL divergence is 0.

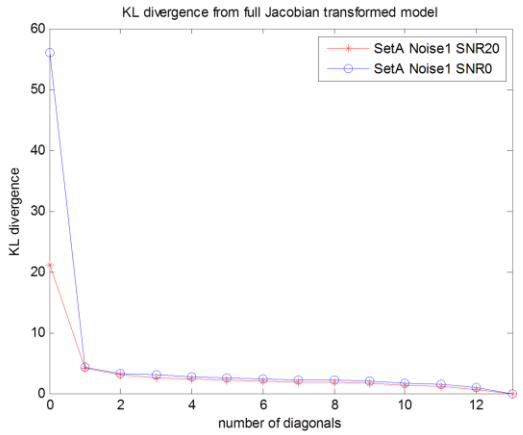

Figure 2: *Average KL divergence between Gaussians adapted using an approximate Jacobian and Gaussians adapted using the full Jacobian for the 20 dB and 0 dB SNR conditions of Aurora 2 test set A with noise type 1 (subway noise).*

We have the following observations from Figure 2:

1) The distortion introduced by setting the off-diagonal elements in Jacobian transform is much smaller than the KL divergence between the clean model and full-Jacobian-transformed model. Therefore, the diagonal approximation of Jacobian matrices is reasonable.
2) Setting the off-diagonal elements in the Jacobian matrix to zero in high SNR conditions brings less distortion than in low SNR conditions. This is consistent with the observation in Figure 1, where the diagonal elements were more dominant in high SNR conditions.

We next compared the recognition accuracies obtained by the different VTS algorithms. Table 2 reports the average of accuracy of all three test sets. Following the standard practice for Aurora 2 [12], the overall average accuracy shown in the last row is computed from SNRs between 0 and 20 dB. In high to moderate SNR conditions (from clean to 5 dB), DJVTS has comparable accuracy to standard MVTS. This validates our observation that at these SNRs, the diagonal elements dominate the Jacobian matrices, and therefore the diagonal approximation is accurate. At SNRs below 5 dB, the approximation is less accurate, and there is an obvious gap between standard MVTS and DJVTS.

We also compared the performance of DJVTS to VTS feature enhancement (FVTS) studied in [10]. Both methods reduce the computational cost of VTS, but with different approaches. FVTS operates using a small GMM with fewer Gaussians than in the HMM to save computation. If M is the number of Gaussians in the front-end GMM in FVTS, the costs of computing the Jacobian matrices, adapting model parameters, and updating the distortion parameters are $\mathcal{O}(MD^3)$, $\mathcal{O}(MD^2)$, and $\mathcal{O}(MD^3)$, respectively. If $MD > N$, then the cost of the feature VTS is actually larger than the cost of DJVTS. In the experiments reported in [10], N=3628, $M = 552$, and $D = 13$. In this case, DJVTS has both better accuracy and lower computational cost than feature VTS.

5. Conclusions

In this paper, we have presented an efficient VTS adaptation algorithm using a diagonal Jacobian approximation. We observed that Jacobian matrix used in the VTS approximation is largely dominated by its diagonal elements, and we therefore proposed to approximate it as a diagonal matrix. We showed this diagonal approximation introduces very small modeling error to the true VTS-adapted model in terms of KL divergence. Using the diagonal Jacobian approximation significantly reduces the computational cost of all the three major components of standard VTS model adaptation, with reductions of $\mathcal{O}(D)$ for Jacobian calculation, $\mathcal{O}(D)$ for most parts of model adaptation, and $\mathcal{O}(D^2)$ for re-estimation of the distortion model parameters. In the experimental evaluation on the Aurora 2 task, the proposed DJVTS has comparable accuracy to standard VTS at SNRs as low as 5 dB. The proposed DJVTS method also outperforms VTS-based feature enhancement at all SNRs, and in some cases may have lower computational cost.

Table 2: *Recognition results for the baseline system and for three VTS setups with clean-trained complex backend HMMs.*

Accuracy	Baseline	Standard MVTS	DJVTS	Feature VTS [10]
Clean	99.48	99.58	99.59	99.59
20 dB SNR	91.38	99.16	99.21	99.10
15 dB SNR	76.99	98.40	98.51	98.38
10 dB SNR	52.98	96.56	96.58	95.90
5 dB SNR	27.69	91.08	90.85	88.74
0 dB SNR	12.81	75.57	71.92	66.75
-5 dB SNR	8.56	40.07	34.41	32.63
Average	50.64	92.18	91.61	89.71

6. References

[1] Kim, D. Y., Un, C. K., Kim, N. S., "Speech recognition in noisy environments using first order vector Taylor series," *Speech Communication*, vol. 24, pp. 39-49, 1998.

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

[3] Acero, A., Deng, L., Kristjansson, T., Zhang, J., "HMM adaptation using vector Taylor series for noisy speech recognition," *Proc. ICSLP*, vol.3, pp. 869-872, 2000.

[4] Li, J., Deng, L., Yu, D., Gong, Y., and Acero, A., "High-performance HMM adaptation with joint compensation of additive and convolutive distortions," *Proc. IEEE ASRU*, 2007.

[5] Li, J., Deng, L., Yu, D., Gong, Y., and Acero, A., "A unified framework of HMM adaptation with joint compensation of additive and convolutive distortions," Computer Speech and Language, no. 3, vol. 23, Elsevier, 2009.

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

[7] Xu, H., Gales, M. J. F., and Chin. K. K., "Improving joint uncertainty decoding performance by predictive methods for noise robust speech recognition," *Proc. ASRU*, 2009.

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

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

[10] Li, J., Seltzer, M. L., and Gong, Y., "Improvements to VTS feature enhancement," *Proc. ICASSP*, 2012.

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

[12] Macho, D, et al., "Evaluation of a noise-robust DSR front-end on Aurora databases," *Proc. ICSLP*, pp. 17–20, 2002.