ACOUSTIC MODEL ADAPTATION VIA LINEAR SPLINE INTERPOLATION FOR ROBUST SPEECH RECOGNITION

Michael L. Seltzer, Alex Acero

Microsoft Research
Redmond, WA USA
{mseltzer,alexac}@microsoft.com

Kaustubh Kalgaonkar *

Georgia Institute of Technology
Atlanta, GA USA
kaustubh@ece.gatech.edu

ABSTRACT

We recently proposed a new algorithm to perform acoustic model adaptation to noisy environments called Linear Spline Interpolation (LSI). In this method, the nonlinear relationship between clean and noisy speech features is modeled using linear spline regression. Linear spline parameters that minimize the error the between the predicted noisy features and the actual noisy features are learned from training data. A variance associated with each spline segment captures the uncertainty in the assumed model. In this work, we extend the LSI algorithm in two ways. First, the adaptation scheme is extended to compensate for the presence of linear channel distortion. Second, we show how the noise and channel parameters can be updated during decoding in an unsupervised manner within the LSI framework. Using LSI, we obtain an average relative improvement in word error rate of 10.8% over VTS adaptation on the Aurora 2 task with improvements of 15-18% at SNRs between 10 and 15 dB.

Index Terms— robust speech recognition, model adaptation

1. INTRODUCTION

Acoustic model adaptation has been proposed as a method of improving speech recognition performance in noisy environments. Some adaptation methods operate in a data-driven manner, but better performance is typically obtained by algorithms that utilize the known relationship between clean speech, noise, and noisy speech. However, because this relationship is nonlinear in the feature domain, the best way to exploit this relationship is an open question.

Several different methods for handling this nonlinearity have been proposed. For example, in data-driven Parallel Model Combination, Monte Carlo sampling is used to generate samples from the constituent speech and noise distributions which are then used to estimate the parameters of the resulting noisy speech distribution [1]. In Vector Taylor Series (VTS) adaptation, e.g. [2], the nonlinear function that describes noisy speech features as a function of the clean speech and noise features is linearized around expansion points defined by the speech and noise models. In [3], an Unscented Transform is used to estimate the noisy speech distribution using a small number of speech and noise sample points.

We recently introduced a novel HMM adaptation scheme called Linear Spline Interpolation (LSI) [4]. In LSI, the relationship between the *a priori* and *a posteriori* SNRs in the log mel spectral domain is modeled using linear spline regression. Adaptation is performed by linearly transforming the *a priori* SNR to obtain an estimate of the *a posteriori* SNR . This transformation is determined by interpolating the parameters of the linear spline. Finally, the noisy speech distribution can be determined from the distribution of the *a posteriori* SNR .

The proposed algorithm has two key advantages. First, the spline parameters are learned from training data. Unlike VTS, the linearization is not restricted to be tangent to the nonlinear function that defines the relationship between clean and noisy speech. Rather, the algorithm can find any set of spline parameters that minimize the error between the predicted and actual noisy speech features. Second, because each line segment has an associated variance, we can capture the uncertainty due to the phase asynchrony between the clean speech and the additive noise [5]. In almost all other model adaptation schemes, this source of uncertainty is ignored.

In this paper, we improve the Linear Spline Interpolation algorithm in two ways. First, we extend the formulation to compensate for linear channel distortion, i.e. spectral tilt, in addition to additive noise. Second, we derive the update equations required to perform unsupervised online re-estimation of the noise and channel parameters during decoding. As the experimental results show, both of these contribute to significantly improved speech recognition accuracy.

2. TRANSFORMING PRIOR SNR TO POSTERIOR SNR

If x, n, h, and y are the log mel spectral representations of the clean speech, noise, channel and noisy speech, respectively, then the noisy log mel spectrum y can be expressed as

$$y = n + \log(1 + e^{(x+h-n)} + 2\alpha e^{(x+h-n)/2}) \quad (1)$$

where α is a random variable that represents the relative phase between the clean speech and the noise [5]. Because the expected value of $\alpha = 0$, most model adaptation and feature enhancement algorithms in the literature ignore its effect and operate on the simplified expression $y = n + \log(1 + e^{(x+h-n)})$. If we define $\tilde{x} = x + h$ as the (possibly) filtered version of clean speech, this expression also shows the relationship between the *a priori* SNR $u = \tilde{x} - n$ and the *a posteriori* SNR $v = y - n$ in the log mel spectral domain:

$$v = \log(1 + e^{u}) \quad (2)$$

Figure 1 shows a two-dimensional histogram of the *a priori* SNR versus the *a posteriori* SNR for the 12th log mel spectral component, derived from the clean and multi-condition training data of the Aurora 2 corpus. As the figure shows, the mode of the data lies on the line defined by (2). However, the data has significant variance around this mode as a result of the phase asynchrony between the clean speech and the noise. By modeling this variance explicitly, Linear Spline Interpolation can achieve more accurate model adaptation.

*A portion of this work was performed while the author was an intern at Microsoft Research

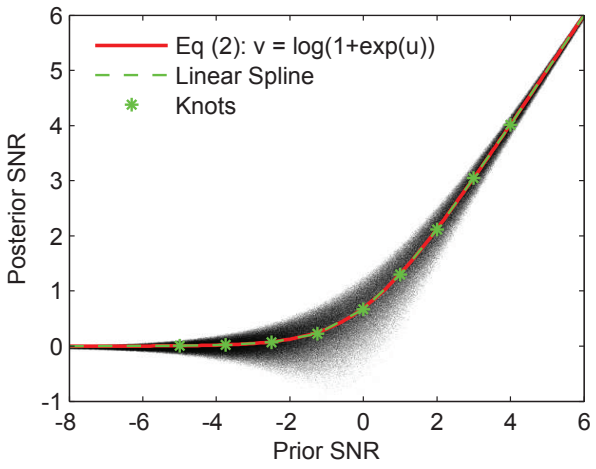

Fig. 1. 2-D histogram of u vs. v for the 12th log mel coefficient, showing the variance of the data. The mode of the data given by (2) and trained linear spline with 10 segments are also shown.

3. LINEAR SPLINE INTERPOLATION

3.1. Linear spline regression

In linear spline regression, pairs of data (x, y) are modeled by a series of segments, where each segment is modeled using linear regression. The regression parameters are computed under the constraint that neighboring regression lines must intersect at the segment boundaries, called *knots*.

In the proposed algorithm, we model the relationship in (2) by a linear spline regression composed of K segments, where the kth segment is defined as

$$v = a_k u + b_k + \epsilon_k, \qquad \forall\, U_{k-1} < u \leq U_k \tag{3}$$

where a_k, b_k, and ϵ_k are the slope, y-intercept, and error of the kth line segment, respectively, and U_{k-1} and U_k are the knots that define the segment boundaries. In each segment, the error ϵ_k is modeled as a zero-mean Gaussian with variance $\sigma^2_{\epsilon_k}$. To find the set of spline parameters given a set of knot locations, we minimize

$$\sum_{n=1}^{N_k} \left\{a_k u_n + b_k - \ln(1 + e^{u_n})\right\}^2, \; U_{k-1} < u_n \leq U_k \tag{4}$$

where N_k is the number of samples that lie in the kth segment. Note that we are minimizing the distance to the mode of the data given by (2). The parameters that minimize (4) for all K segments and satisfy the adjacency constraints are found by solving a system of linear equations, as described in [6]. As shown in Figure 1, the linear spline regression closely approximates the nonlinear function in (2).

3.2. Minimum Mean Square Error Estimate of Noisy Speech

We can use the linear spline to construct an MMSE estimate of y given x and n. To do so, we first construct an MMSE estimate of the *a posteriori* SNR v given the *a priori* SNR u. This estimate is a weighted sum of the means of the segment-conditional posterior distribution $p(v|u,k) = \mathcal{N}(v; a_k u + b_k, \sigma^2_{\epsilon_k})$, and can can be written

$$\hat{v} = \sum_k w_k \int v p(v|u,k) dv = \sum_k w_k (a_k u + b_k). \tag{5}$$

where w_k represents the probability that u lies in the kth line segment. We compute w_k as

$$w_k = p(k) = \int p(k,u) du = \int p(k|u) p(u) du \tag{6}$$

where $p(k|u) = \delta(u \in (U_{k-1}, U_k])$, i.e. $p(k|u) = 1$ if u is in the kth segment and 0 otherwise. Of course, if u is known, then $p(u)$ is also a delta function at the value of u, and $w_k = 1$ for the segment containing u, and $w_k = 0$ for all other segments.

By substituting the definitions of u and v into (5) and rearranging terms, the MMSE estimate of y can be computed as

$$\hat{y} = (1 - \textstyle\sum_k w_k a_k) n + (\sum_k w_k a_k)(x + h) + \sum_k w_k b_k \tag{7}$$

4. HMM ADAPTATION USING LINEAR SPLINE INTERPOLATION

Our goal is to estimate the parameters of the noisy speech distribution $p(y)$. The mean of y is computed by applying the expectation operator to both sides of (7). The variance can then be computed from the second moment $E[y^2]$ and the estimate of the mean of y. This gives the following estimates of the mean and variance of $p(y)$

$$\mu_y = (1 - \sum_k w_k a_k)\mu_n + (\sum_k w_k a_k)(\mu_x + \mu_h) + \sum_k w_k b_k \tag{8}$$

$$\sigma^2_y = (1 - \sum_k w_k a_k)^2 \sigma^2_n + (\sum_k w_k a_k)^2 \sigma^2_x + \sum_k w_k^2 \sigma^2_{\epsilon_k} \tag{9}$$

In the previous section, it was assumed that u was known. As a result, w_k was 1 for the segment that bounded u and 0 otherwise. In the model domain, u is unknown, and therefore w_k is computed based on the probability distribution of u. If we assume that x and n are independent Gaussian random variables then u is also Gaussian with mean $\mu_u = \mu_x - \mu_n$ and variance $\sigma^2_u = \sigma^2_x + \sigma^2_n$. Using this distribution for u and (6), we compute w_k as

$$w_k = \int_{U_k - 1}^{U_k} p(u) du = \Phi(U_k; \mu_u, \sigma^2_u) - \Phi(U_{k-1}; \mu_u, \sigma^2_u) \tag{10}$$

where Φ is the continuous density function (CDF) of a Gaussian distribution, and $\{U_{k-1}, U_k\}$ are the knots of the kth segment.

The adaptation formulae in (8) and (9) are valid for log mel spectral components. To transform these to the cepstral domain, we define the following terms

$$\boldsymbol{A} = diag\left(\textstyle\sum_k w_{1k} a_{1k}, \; \cdots, \sum_k w_{Lk} a_{Lk}\right) \tag{11}$$

$$\boldsymbol{b} = \left[\textstyle\sum_k w_{1k} b_{1k}, \; \cdots, \sum_k w_{Lk} b_{Lk}\right]^T \tag{12}$$

$$\boldsymbol{\Sigma}_\epsilon = diag\left(\textstyle\sum_k w^2_{1k} \sigma^2_{\epsilon_{1k}}, \; \cdots, \sum_k w^2_{Lk} \sigma^2_{\epsilon_{Lk}}\right) \tag{13}$$

where $\left\{a_{lk}, b_{lk}, \sigma^2_{\epsilon_{lk}}\right\}$ are the spline parameters for the kth spline segment of the lth log mel coefficient. We additionally define $\boldsymbol{e} = \boldsymbol{C}\boldsymbol{b}$, $\boldsymbol{F} = \boldsymbol{C}\boldsymbol{A}\boldsymbol{D}$ and $\boldsymbol{G} = \boldsymbol{I} - \boldsymbol{F}$, where $\boldsymbol{C}$ is the truncated DCT and $\boldsymbol{D}$ is the pseudo-inverse of $\boldsymbol{C}$. The cepstral model parameters can now be transformed as

$$\boldsymbol{\mu}_y = \boldsymbol{F}\boldsymbol{\mu}_x + \boldsymbol{G}\boldsymbol{\mu}_n + \boldsymbol{e}_s \tag{14}$$

$$\boldsymbol{\Sigma}_y = \boldsymbol{F}\boldsymbol{\Sigma}_x\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{\Sigma}_n\boldsymbol{G}^T + \boldsymbol{C}\Sigma_\epsilon\boldsymbol{C}^T \tag{15}$$

Note that even though $\boldsymbol{\Sigma}_y$ is a full matrix, we assume it is diagonal for decoding purposes.

The adaptation equations for the dynamic model parameters are similar to the static parameters. They have been omitted for space considerations but can be found in [4].

5. UNSUPERVISED NOISE AND CHANNEL RE-ESTIMATION USING LSI

In [4], LSI was used to perform HMM adaptation for a fixed estimate of the noise parameters and no channel distortion ($\boldsymbol{\mu}_h = 0$). We now show how the noise and channel parameters can be re-estimated in an unsupervised manner using a Generalized EM approach. We start with the following auxiliary function

$$Q(\lambda, \hat{\lambda}) = \sum_{t,s} \gamma_{ts} \log(p(\boldsymbol{y}_t|s, m, \lambda)) \tag{16}$$

where γ_{ts} is the posterior probability of Gaussian component s occurring at frame t given the observation sequence and $p(y_t|s, \lambda) = \mathcal{N}(\boldsymbol{y}_t, \boldsymbol{\mu}_{y,s}, \boldsymbol{\Sigma}_{y,s})$ is the likelihood of the observation under the adapted LSI model.

5.1. Re-estimation of the noise and channel means

The re-estimation formulae for the noise mean can be determined by taking the derivative of (16) with respect to $\boldsymbol{\mu}_n$ and setting the result equal to 0. Solving for $\boldsymbol{\mu}_n$, we obtain

$$\boldsymbol{\mu}_n = \left(\sum_{t,s} \gamma_{ts} \boldsymbol{G}_s^T \boldsymbol{\Sigma}_{y,s}^{-1} \boldsymbol{G}_s \right)^{-1} \times \left(\sum_{t,s} \gamma_{ts} \boldsymbol{G}_s^T \boldsymbol{\Sigma}_{y,s}^{-1} (\boldsymbol{y}_t - \boldsymbol{F}_s(\boldsymbol{\mu}_{x,s} + \boldsymbol{\mu}_h) - \boldsymbol{e}_s) \right) \tag{17}$$

Note that the transformation parameters now have a subscript s to indicate they are a function of the Gaussian component. The update equation for the channel mean $\boldsymbol{\mu}_h$ can be similarly computed as

$$\boldsymbol{\mu}_h = \left(\sum_{t,s} \gamma_{ts} \boldsymbol{F}_s^T \boldsymbol{\Sigma}_{y,s}^{-1} \boldsymbol{F}_s \right)^{-1} \times \left(\sum_{t,s} \gamma_{ts} \boldsymbol{F}_s^T \boldsymbol{\Sigma}_{y,s}^{-1} (\boldsymbol{y}_t - \boldsymbol{F}_s \boldsymbol{\mu}_{x,s} - \boldsymbol{G}_s \boldsymbol{\mu}_n - \boldsymbol{e}_s) \right) \tag{18}$$

The re-estimation formulae for the means of the dynamic noise and channel parameters can be similarly computed. However, in this work we assume the the channel is fixed and the noise is stationary. As a result, the means of the dynamic noise and channel parameters are simply set equal to zero.

5.2. Re-estimation of the noise variances

Unfortunately, there is no closed-form solution for the noise variance update. Therefore, we update the variance iteratively using Newton's method. The new estimate of the variance is computed as

$$\boldsymbol{\Sigma}_n^{\text{new}} = \boldsymbol{\Sigma}_n - [\mathcal{H}(\boldsymbol{\Sigma}_n)]^{-1} [\nabla Q(\boldsymbol{\Sigma}_n)] \tag{19}$$

where $\mathcal{H}(\boldsymbol{\Sigma}_n)$ is the Hessian matrix with elements defined as

$$\mathcal{H}_{ij}(\boldsymbol{\Sigma}_n) = \frac{\partial^2 Q}{\partial \sigma_n^2(i) \partial \sigma_n^2(j)} \tag{20}$$

Because the variances are not guaranteed to remain non-negative, we optimize $\tilde{\boldsymbol{\Sigma}}_n = \log(\boldsymbol{\Sigma}_n)$ in practice. Due to space constraints, we omit the derivation, but we give the expressions for the terms of the gradient and Hessian below.

$$\frac{\partial Q}{\partial \tilde{\sigma}_n^2(i)} = -\frac{1}{2} \sigma_n^2(i) \sum_{t,s} \gamma_{ts} \times \sum_d \left\{ \frac{G_s(d,i)^2}{\sigma_{y,s}^2(d)} \times \left(1 - \frac{(y_t(d) - \mu_{y,s}(d))^2}{\sigma_{y,s}^2(d)} \right) \right\} \tag{21}$$

$$\frac{\partial^2 Q}{\partial \tilde{\sigma}_n^2(i) \partial \tilde{\sigma}_n^2(j)} = \frac{1}{2} \sigma_n^2(i) \sigma_n^2(j) \sum_{t,s} \gamma_{ts} \times \sum_d \left\{ \frac{G_s(d,i)^2 G_s(d,j)^2}{[\sigma_{y,s}^2(d)]^2} \left(1 - 2 \frac{(y_t(d) - \mu_{y,s}(d))^2}{\sigma_{y,s}^2(d)} \right) - \delta(i-j) \sum_d \frac{G_s(d,i)^2}{\sigma_{y,s}^2(d)} \left(1 - \frac{(y_t(d) - \mu_{y,s}(d))^2}{\sigma_{y,s}^2(d)} \right) \right\} \tag{22}$$

This approach is also used to update the variances of the dynamic noise parameters ($\boldsymbol{\Sigma}_{\Delta n}, \boldsymbol{\Sigma}_{\Delta\Delta n}$). We assume that the static, delta, and delta-delta components are independent, so the Hessian matrices for each set of parameters can be computed independently.

5.3. Algorithm Implementation

We now summarize the sequence of steps involved in performing model adaptation using LSI.

1. Read in the noisy utterance
2. Initialize $\boldsymbol{\mu}_h = 0$ and compute sample estimates of $\{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n, \boldsymbol{\Sigma}_{\Delta n}, \boldsymbol{\Sigma}_{\Delta\Delta n}\}$ from the first and last N frames of the utterance.
3. For each Gaussian, compute the spline weights $\{w_k\}$ and transformation parameters $\{\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{e}, \boldsymbol{\Sigma}_\epsilon\}$.
4. Adapt the HMM parameters and decode the utterance.
5. Using the hypothesized transcription, compute posterior probabilities γ_{st} and re-estimate the channel and noise parameters.
6. For each Gaussian, recompute the spline weights and transformation parameters and adapt the HMM parameters.
7. Decode the utterance and output the transcription.

This sequence of steps constitutes a single iteration of Generalized EM for updating the noise and channel parameters. If multiple iterations are desired, an inner loop of steps 5 and 6 can be performed. In our implementation, three iterations of Newton's method were performed to update the noise variances during the M-step.

6. EXPERIMENTS

In order to evaluate the performance of the proposed LSI model adaptation technique, a series of experiments were performed on the Aurora 2 corpus [7]. Aurora 2 consists of data degraded with eight types of noise at SNRs between 0 dB and 20 dB. Evaluation is performed using three test sets that contain noise types seen in the training data (Set A), unseen in the training data (Set B), and additive noise plus channel distortion (Set C).

The acoustic models were trained from the clean training set using HTK with the standard "complex back end" Aurora 2 recipe. An HMM with 16 states per digit and 20 Gaussians per state is created for each digit as a whole word. There is a three state silence model with 36 Gaussians per state and a one state short pause model tied to the middle state of silence. Standard 39-dimensional MFCC features consisting of 13 static, delta, and delta-delta features were used and C0 was used instead of log energy. Noise is assumed to be stationary and Gaussian with a diagonal covariance. The baseline word accuracy with no compensation is 62.6%.

A linear spline was trained for each mel component using the clean and multi-condition training data from Aurora 2. Each spline consisted of 36 segments. This number was shown to have good performance in previous work [4]. The knot locations were chosen empirically. Knots were more densely placed at values of u between

SNR (dB)	Set A		Set B		Set C		Avg	
	VTS	LSI	VTS	LSI	VTS	LSI	VTS	LSI
∞	99.6	99.6	99.6	99.6	99.6	99.6	99.6	99.6
20	99.0	99.1	99.2	99.2	99.0	99.0	99.1	99.1
15	97.9	98.3	98.2	98.5	98.0	98.2	98.0	98.4
10	94.8	95.8	95.4	96.2	94.8	95.9	95.0	95.9
5	87.2	88.7	88.1	89.2	87.2	88.2	87.5	88.7
0	68.6	71.0	70.2	71.8	70.6	71.6	69.8	71.5
-5	31.9	38.6	33.7	38.5	39.2	40.9	34.9	39.3
Avg	89.5	90.6	90.2	91.0	89.9	90.6	89.9	90.7

Table 1. Accuracy obtained by LSI and VTS as a function of SNR.

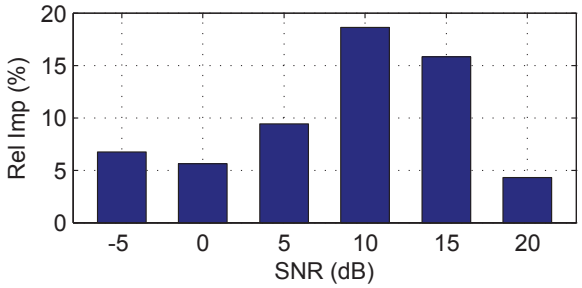

Fig. 2. Relative improvement in word error rate of LSI over VTS adaptation as a function of SNR

-5 dB and 5 dB based on the observation that the variance of $p(v|u)$ changes more quickly at SNRs near 0 dB.

We compared the performance of the proposed LSI adaptation method to VTS adaptation [2]. In both LSI and VTS, the noise and channel parameters were initialized in the same way and the exact same optimization procedure was used for parameter re-estimation. The static, delta, and delta-delta means and variances were adapted in both cases.

The first experiment we performed used a single iteration of the noise and channel parameter re-estimation. The results are shown in Table 1 as a function of the SNR for each test set. Note that '∞' indicates clean speech and that the average accuracy in the last row only includes SNRs between 0 dB and 20 dB. As the table shows, the proposed LSI algorithm outperforms VTS at every SNR with the exception of clean speech in which the performance is identical. Note that both VTS and LSI outperform the ETSI Advanced Front End, which achieves an average accuracy of 88.6% on this task.

The relative improvement in word error rate of LSI adaptation over VTS is shown in Figure 2. As the figure shows, LSI provides the most gain at the moderate SNRs (10-15 dB). This is significant because it is speech at these SNRs that the our algorithm is directly trying to improve. In speech with high SNR, most of the spectral components will lie in the upper right portion of Figure 1, whereas in speech with low SNR, many of the components will lie in the lower left corner of Figure 1. In both these regions, the function is linear and the variance is low. However, at moderate SNRs, more spectral components are concentrated around 0 dB which is the portion of the curve that has the highest variance and is the most nonlinear. We believe that the improvements obtained at these SNRs are due to the proposed algorithm's ability to 1) explicitly model the variance of the transformation from clean to noisy speech, and 2) produce a more accurate linearization compared to VTS whose linearization is forced to be tangent to the function in (2).

Iter	Accuracy (%)		Relative Imp (%)
	VTS	LSI	
0	88.3	89.2	8.2
1	89.9	90.7	8.6
2	90.2	91.0	7.9

Table 2. Accuracy obtained by LSI and VTS as a function of iterations of noise and channel parameter re-estimation

Finally, Table 2 shows the effect of additional iterations of re-estimation of the noise and channel parameters. As the table shows, additional accuracy is obtained for both VTS and LSI, and the relative improvement of LSI over VTS is maintained at each iteration.

7. CONCLUSIONS

In this paper, we proposed a novel method for acoustic model adaptation called Linear Spline Interpolation. In LSI, the relationship between the *a priori* and *a posteriori* SNRs in the log mel spectral domain is modeled using linear spline regression. The spline is learned from training data and the resulting parameters are interpolated at runtime to adapt the HMM parameters in order to compensate for additive noise and linear filtering. We demonstrated how the noise and channel parameters can be updated in an unsupervised manner during decoding. Using the proposed LSI algorithm, we obtained a significant improvement over VTS adaptation at all SNRs with a maximum gain of 18.6% at 10 dB. In the future, we plan to investigate improvements to the training of the linear spline, including automatic selection of the knot locations and HMM-based learning of the spline parameters.

8. REFERENCES

[1] M. J. F. Gales and S. J. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Trans. on Speech and Audio Proc.*, vol. 4, pp. 352–359, 1996.

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

[3] Y. Hu and Q. Huo, "An HMM compensation approach using unscented transformation for noisy speech recognition," in *Proc. ISCSLP*, nov 2006, pp. 346–357.

[4] K. Kalgaonkar, M. L. Seltzer, and A. Acero, "Noise robust model adaptation using linear spline interpolation," in *Proc. of ASRU*, Trento, Italy, 2009.

[5] L. Deng, J. Droppo, and A. Acero, "Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise," *IEEE Trans. on Speech and Audio Proc.*, vol. 12, no. 2, pp. 133–143, March 2004.

[6] J. E. Ertel and E. B. Fowlkes, "Some algorithms for linear spline and piecewise multiple linear regression," *Journal of the Amer. Stat. Assc.*, vol. 71, no. 355, pp. 640–648, Sept. 1976.

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