JOINT UNCERTAINTY DECODING WITH THE SECOND ORDER APPROXIMATION FOR NOISE ROBUST SPEECH RECOGNITION

Haitian Xu, K.K. Chin

Speech Technology Group, Toshiba Research Europe Ltd.
Cambridge Research Laboratory, Cambridge, UK
{haitian.xu, kkchin}@crl.toshiba.co.uk

ABSTRACT

Joint uncertainty decoding has recently achieved promising results by integrating the front-end uncertainty into the back-end in a mathematically consistent framework. In this paper, joint uncertainty decoding is compared with the widely used vector Taylor series (VTS). We show that the two methods are identical except that joint uncertainty decoding applies the Taylor expansion on each regression class whereas VTS applies it to each HMM mixture. The relatively rougher expansion points used in joint uncertainty decoding make it computationally cheaper than VTS but inevitably worse on recognition accuracy. To overcome this drawback, this paper proposes an improved joint uncertainty decoding algorithm which employs second-order Taylor expansion on each regression class in order to reduce the expansion errors. Special considerations are further given to limit the overall computational cost by adopting different number of regression classes for different orders in the Taylor expansion. Experiments on the Aurora 2 database show that the proposed method is able to beat VTS on recognition accuracy and computational cost with relative improvement up to 6% and 60%, respectively.

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. This relatively low robustness against environmental noise makes it difficult to deploy ASR technology in real applications.

One way to tackle this problem is to adapt previously trained clean speech hidden Markov models (HMM) to the encountered noisy environment. One of the popular techniques is vector Taylor series (VTS) [1] [2] which applies linear approximation on the non-linear noise corruption for each HMM mixture by first-order Taylor series. Although promising results have been achieved [3], the computational cost of VTS is relatively high since the Taylor series expansion needs to be calculated for each mixture in the HMM.

Recently, another model adaptation technique, joint uncertainty decoding (JUD), was introduced [4]. By modelling the relationship between clean and noisy speech with their joint distribution, this method adapts the HMM in a mathematically consistent framework. To make it efficient, the estimation of the joint distribution is normally simplified by using the first-order Taylor series expansion [5]. With this simplification, JUD has proved to be much faster than VTS but shows slightly worse recognition accuracy [6].

In this paper, we compare JUD and VTS, and prove that the two methods are equivalent except that JUD employs fewer - therefore rougher - Taylor expansion points than VTS. This makes JUD more flexible and explains why it has less computational cost but cannot beat VTS on recognition accuracy. In order to further boost the JUD performance, this paper proposes to employ second order Taylor expansion on JUD. This is sensible for two reasons. On the one hand, higher order Taylor expansion provides better approximation than first order expansion especially when the Taylor expansion points are rough. On the other hand, although directly applying higher order approximation is computationally prohibitive for VTS, it could be very efficient for JUD because fewer expansion points are involved in JUD. To further minimise the computational cost, a more flexible Taylor expansion scheme for JUD is introduced by using different number of expansion points for different Taylor series orders. On the Aurora 2 task [7], the proposed technique achieves 6% relative word error rate reduction compared to VTS and has 60% improvement on computational cost.

The remainder of this paper is as follows: section 2 gives an overview of VTS and JUD techniques and compares them in a theoretical point of view; section 3 introduces JUD with the second order approximation; in section 4, experimental results on Aurora 2 are presented and conclusions are finally drawn in section 5.

2. VECTOR TAYLOR SERIES AND JOINT UNCERTAINTY DECODING

2.1. Vector Taylor Series

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

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

where C denotes the discrete cosine transformation matrix, n, h, x and y the static features for additive noise, convolutional noise, clean speech and noisy speech, respectively. Given a Taylor expansion point (x_e, n_e, h_e), the above non-linear relationship can be linearly approximated by the first-order Taylor series as:

$$\begin{aligned} y \approx x_e + h_e + g(x_e, n_e, h_e) + W(x - x_e) \\ +(I - W)g(x_e, n_e, h_e)(n - n_e) + W(h - h_e) \\ W = I + \bigtriangledown_x g(x_e, n_e, h_e) \end{aligned} \quad (2)$$

where I is the identity matrix.

Applying Eq.(2) for each mixture of the HMM, we can adapt the clean speech HMM to the noisy environment. For the mth mixture, we denote the mean and variance for clean speech with their static, delta and delta-delta parts as $\Lambda_x^m = (\mu_x^m, \triangle\mu_x^m, \triangle\triangle\mu_x^m)$ and $\Xi_x^m = (\Sigma_x^m, \triangle\Sigma_x^m, \triangle\triangle\Sigma_x^m)$, respectively. As shown in [2], such

an adaptation process takes place by using the static part of the mixture mean and noise mean (μ_x^m, μ_n, μ_h) as the expansion point and the first-order derivative W becomes mixture-dependent (denoted as W_m). The adapted static parts μ_y^m and Σ_y^m are obtained as

$$\mu_y^m \approx \mu_x^m + \mu_h + g(\mu_x^m, \mu_n, \mu_h) \quad (3)$$

$$\Sigma_y^m \approx W_m \Sigma_x^m W_m^T + (I - W_m)\Sigma_n (I - W_m)^T \quad (4)$$

It is reasonable to consider the delta and delta-delta features as the first and second order derivative of the static features over time [8]. Thus, we can adapt the dynamic parts:

$$\triangle\mu_y^m \approx W_m \triangle\mu_x^m \quad (5)$$

$$\triangle\triangle\mu_y^m \approx W_m \triangle\triangle\mu_x^m \quad (6)$$

$$\triangle\Sigma_y^m \approx W_m \triangle\Sigma_x^m W_m^T + (I - W_m)\triangle\Sigma_n (I - W_m)^T \quad (7)$$

$$\triangle\triangle\Sigma_y^m \approx W_m \triangle\triangle\mu_x^m W_m^T + (I - W_m)\triangle\triangle\Sigma_n (I - W_m)^T \quad (8)$$

2.2. Joint Uncertainty Decoding

In [4], JUD is introduced in a mathematically consistent framework. Given the full noisy feature Y which includes the static, delta and delta-delta parts, it calculates the output probability for the mixture m as follows:

$$p(Y|m) = |A_r| N(A_r Y + b_r; \Lambda_x^m, \Xi_x^m + \Xi_b^r) \quad (9)$$

Assuming mixture m belongs to the rth regression class, A_r, b_r and Ξ_b^r in Eq.(9) are the JUD transforms related to the same regression class and defined as follows:

$$A_r = \Xi_x^r (\Xi_{yx}^r)^{-1}, b_r = \Lambda_x^r - A_r \Lambda_y^r$$
$$\Xi_b^r = A_r \Xi_y^r {A_r}^T - \Xi_x^r$$

where Λ_x^r, Ξ_x^r, Λ_y^r and Ξ_y^r are the mean and covariance respectively for clean and noisy speech in regression class r, and Ξ_{yx}^r is the cross-covariance matrix. The calculation of Ξ_{yx}^r turns out to be very costly and one solution is to use the Taylor expansion in Eq.(1) on it [5]. A_r then becomes

$$A_r = \begin{pmatrix} W_r^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & W_r^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & W_r^{-1} \end{pmatrix} \quad (10)$$

By applying Eq.(10) and after some mathematical manipulations, Eq.(9) could be reformulated as:

$$p(Y|m) = N(Y; \Lambda_y^m, \Xi_y^m) \quad (11)$$

This effectively makes JUD a pure model adaptation method where the noisy HMM parameters are calculated as follows.

$$\mu_y^m = \mu_x^r + \mu_h + g(\mu_x^r, \mu_n, \mu_h) + W_r(\mu_x^m - \mu_x^r) \quad (12)$$

$$\triangle\mu_y^m = W_r \triangle\mu_x^m \quad (13)$$

$$\triangle\triangle\mu_y^m = W_r \triangle\triangle\mu_x^m \quad (14)$$

$$\Sigma_y^m = W_r \Sigma_x^m W_r^T + (I - W_r)\Sigma_n (I - W_r)^T \quad (15)$$

$$\triangle\Sigma_y^m = W_r \triangle\Sigma_x^m W_r^T + (I - W_r)\triangle\Sigma_n (I - W_r)^T \quad (16)$$

$$\triangle\triangle\Sigma_y^m = W_r \triangle\triangle\Sigma_x^m W_r^T + (I - W_r)\triangle\triangle\Sigma_n (I - W_r)^T \quad (17)$$

Comparing Eq.(12)-(17) with Eq.(3)-(8), it is obvious that the only difference between JUD and VTS is the selection of the Taylor expansion point. Instead of using the mean of each mixture μ_x^m, JUD applies the expansion over the mean of the regression class μ_x^r. In another word, VTS is a special case of JUD where the number of regression classes equals to the number of mixtures i.e. $\mu_x^r = \mu_x^m$. This has two effects on the performance. First, since the number of regression classes is normally smaller than the number of mixtures, JUD involves much fewer number of derivatives W in the Taylor expansion and therefore is advantageous on computational cost. Second, the expansion point μ_x^r is rougher than the μ_x^m, i.e. statistically farther to the value of real clean speech x. The rougher the expansion point, the larger the Taylor approximation error will be. Thus, JUD is expected to achieve worse recognition accuracy than VTS.

3. JOINT UNCERTAINTY DECODING WITH SECOND ORDER APPROXIMATION

It is generally believed [9] that the higher order Taylor expansion could reduce the Taylor approximation errors, especially when the expansion point is rough. For JUD, this is expected to be helpful for improving the recognition accuracy.

3.1. Second Order Approximation

Given the expansion point (μ_x^r, μ_n, μ_h), the second order Taylor expansion on the feature vector is as follows

$$\begin{aligned} y =& \mu_x^r + \mu_h + g(\mu_x^r, \mu_n, \mu_h) + W_r(x - \mu_x^r) \\ &+ \frac{1}{2} K_r diag\{(x - \mu_x^r)(x - \mu_x^r)^T + (n - \mu_n)(n - \mu_n)^T \\ &- (x - \mu_x^r)(n - \mu_n)^T - (n - \mu_n^r)(x - \mu_x)^T\} \end{aligned} \quad (18)$$

where $diag\{.\}$ outputs the diagonal of the given matrix as a vector and $diag^{-1}\{.\}$ expands a vector into a diagonal matrix. The second order derivative K_r is calculated as

$$\begin{aligned} K_r =& \bigtriangledown_{xx} g(\mu_x^r, \mu_n, \mu_h) \\ =& C diag^{-1}\{\frac{e^{C^{-1}(\mu_n - \mu_x^r - \mu_h)}}{[1 + e^{C^{-1}(\mu_n - \mu_x^r - \mu_h)}]^2}\} C^{-1} C^{-1} \end{aligned}$$

Taking the mean value on both sides of Eq.(18), the new JUD formulae for HMM mean adaptation are obtained as

$$\mu_y^m = \mu_x^r + \mu_h + g(\mu_x^r, \mu_n, \mu_h) + W_r(\mu_x^m - \mu_x^r) + K_r d^m \quad (19)$$

$$\triangle\mu_y^m = W_r \triangle\mu_x^m + K_r d_\triangle^m \quad (20)$$

$$\triangle\triangle\mu_y^m = W_r \triangle\triangle\mu_x^m + K_r d_{\triangle\triangle}^m \quad (21)$$

In the formulae above, d^m, $d_\triangle^m$ and $d_{\triangle\triangle}^m$ are vectors depending on each mixture

$$d^m = \frac{1}{2} diag\{[\Sigma_x^m + \Sigma_n + (\mu_x^m - \mu_x^r)(\mu_x^m - \mu_x^r)^T]\}$$

$$d_\triangle^m = diag\{(\mu_x^m - \mu_x^r)\triangle{\mu_x^m}^T\}$$

$$d_{\triangle\triangle}^m = diag\{(\mu_x^m - \mu_x^r)\triangle\triangle{\mu_x^m}^T + \triangle\Sigma_x^m + \triangle\Sigma_n + \triangle\mu_x^m \triangle{\mu_x^m}^T\}$$

Although a new formula for HMM variance adaptation can also be acquired with Eq.(18), this paper keeps using the first order approximation in Eq.(15)-Eq.(17) on variance parts both for simplicity and to minimise the extra computational cost involved. Applying Eq.(19)-(21) to Eq.(11), JUD with second-order approximation can be written in a form similar to Eq.(9):

$$p(Y|m) = |A_r| N(A_r Y + b_r; \Lambda_x^m + \Lambda_b^m, \Xi_x^m + \Xi_b^r) \quad (22)$$

where we can observe that

$$\Lambda_b^m = \begin{pmatrix} W_r^{-1} K_r d^m \\ W_r^{-1} K_r d_{\triangle}^m \\ W_r^{-1} K_r d_{\triangle\triangle}^m \end{pmatrix} \quad (23)$$

is the only difference between Eq.(22) and Eq.(9).

Clearly, JUD with second-order approximation has several advantages over VTS. First, since the second order Taylor expansion is applied, the JUD performance can be as good as the computationally prohibitive second-order VTS. In other words, the new JUD is able to beat the commonly used first-order VTS on recognition accuracy, which is almost impossible for the original JUD. Second, although the new JUD formula does introduce extra computational cost on the calculation of Λ_b^m per mixture, its overall cost is however still expected to be far less than VTS because the most costly parts, the computation of the W_r and K_r, are only for each regression class. Therefore it is expected that JUD with the second order approximation could beat VTS both on computational cost and recognition accuracy.

3.2. Further Simplification

The g vector and the W_r and K_r matrices are most costly in the second-order approximation. Therefore, reducing the overall number of these matrices and vectors should be helpful for further improving the efficiency. Based on the theory of Taylor expansion, the lower order terms in the Taylor series are more important than the higher order ones. Specifically, g is most important for the Taylor expansion, W_r is less important whereas K_r is least important. This means a slightly less accurate K_r will not result in significant performance change. Thus it is reasonable to employ different number of regression classes for different orders, more for the lower orders and fewer for the higher orders. In this paper we adopted the same number of regression classes for g and W_r and uses a different number of classes for K_r.

In addition, we observed in the experiments that applying the second-order approximation for JUD on the delta-delta feature part brings almost no improvement compared to the static and delta parts. Considering the computational cost, this paper only applies the second order JUD on the static and delta parts and keeps using the first order JUD for the delta-delta part.

4. EXPERIMENTS

Experiments were conducted on the Aurora 2 database [7] 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 used the clean training set to train the models and only test set A and B for the recognition test. Recognition was performed with HTK [10]. Each digit was modelled by 16 HMM states with three mixtures whereas the silence was modelled by 3 states each with 6 mixtures - 546 mixtures in all. The front-end was a 13-dimensional MFCC including the zeroth coefficient with their delta and delta-delta components.

The recognition process was implemented in a two pass mode similar to [3]. Specifically,

1. the initial noise parameters μ_n, Σ_n and μ_h as well as their delta and delta-delta terms were estimated from the first and last 20 frames in each utterance
2. first-order VTS was then applied to adapt the HMM in order to generate an initial recognition hypothesis
3. an Expectation-Maximisation based VTS noise estimation process [6] was adopted to refine the noise parameters based on the initial hypothesis
4. the refined noise parameters are finally fed into VTS or JUD to compensate the HMM and obtain the final recognition results

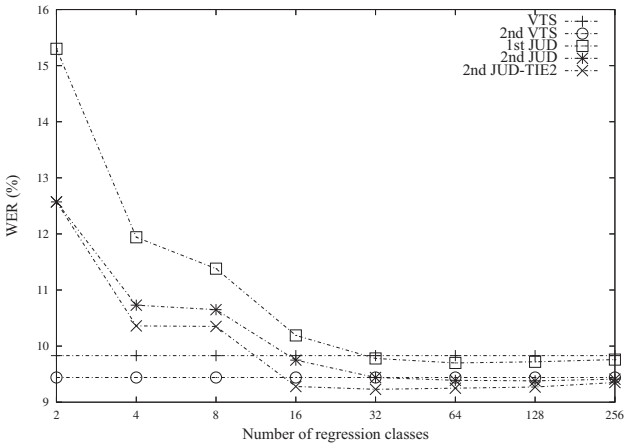

Fig. 1. Averaged WER on Set A with different number of regression classes

#Reg	VTS	2^{nd} VTS	1^{st} JUD	2^{nd} JUD	2^{nd} JUD-TIE2
8	546/546	1092/546	8/8	16/8	10/8
16	546/546	1092/546	16/16	32/16	18/16
32	546/546	1092/546	32/32	64/32	34/32

Table 2. Total number of transforms (#W+#K / #g) involved in the VTS/JUD with 8, 16 or 32 regression classes

With the same noise estimation in step 4, figure 1 gives the average word error rate (WER) on set A for VTS and JUD with different number of regression classes for Taylor expansion. For VTS, results for both the first and second order are provided. For JUD, we also provide the results of a simplified version (denoted as 2^{nd} JUD-TIE2) for the second-order approximation which, as introduced in section 3.2, uses only 2 regression classes for the K_r calculation. With the number of regression classes increasing, it is observed that the performance of JUD with 1st-order approximation gets closer and closer to the first order VTS. By applying the second order Taylor expansion, JUD, either simplified or not, improves the performance consistently. The performance becomes significantly better than the first-order VTS when the number of regression classes is larger than 16. Although not aimed at reducing the WER, the simplified version does achieve a better performance than the non-simplified version on the Aurora 2. The more detailed results shown in table 1 indicate that the simplified 2nd-order JUD, even with 32 regression classes, brings 6.1% relative improvement on WER for set A and 5.6% for set B than VTS. Such an improvement is also consistent across all the noise types.

Table 2 shows the number of transforms i.e. K and W matrices involved in each method. The computational costs for VTS and JUD

#Reg Class	Method	Set A					Set B				
		Subway	Babble	Car	Exhibition	**Ave.**	Restaurant	Street	Airport	Station	**Ave.**
-	Baseline	35.92	52.34	46.60	40.30	**43.79**	49.27	41.21	47.86	47.97	**46.58**
-	VTS	10.76	11.68	7.39	9.48	**9.83**	10.79	9.82	7.60	8.03	**9.06**
-	2^{nd} VTS	10.22	10.98	7.25	9.29	**9.44**	10.14	9.44	7.29	7.70	**8.64**
8	1^{st} JUD	12.28	13.28	9.04	10.99	**11.38**	11.70	11.42	8.68	9.26	**10.27**
	2^{nd} JUD	11.51	11.28	8.44	11.38	**10.65**	10.57	10.58	7.76	8.47	**9.35**
	2^{nd} JUD-TIE2	10.59	10.90	8.76	11.14	**10.35**	10.08	10.42	7.64	8.41	**9.14**
16	1^{st} JUD	11.02	11.73	7.85	10.16	**10.19**	10.99	10.31	7.64	8.29	**9.31**
	2^{nd} JUD	10.47	10.77	7.67	10.10	**9.75**	10.19	9.86	7.31	7.86	**8.81**
	2^{nd} JUD-TIE2	9.74	10.48	7.42	9.48	**9.28**	9.85	9.49	7.09	7.56	**8.50**
32	1^{st} JUD	10.82	11.47	7.43	9.38	**9.78**	10.76	9.90	7.53	8.04	**9.06**
	2^{nd} JUD	10.24	10.68	7.41	9.44	**9.44**	9.97	9.63	7.17	7.70	**8.62**
	2^{nd} JUD-TIE2	9.86	10.61	7.36	9.11	**9.23**	9.93	9.48	7.11	7.69	**8.55**

Table 1. WER (%) averaged over each noise type for different methods with 8, 16 or 32 regression classes

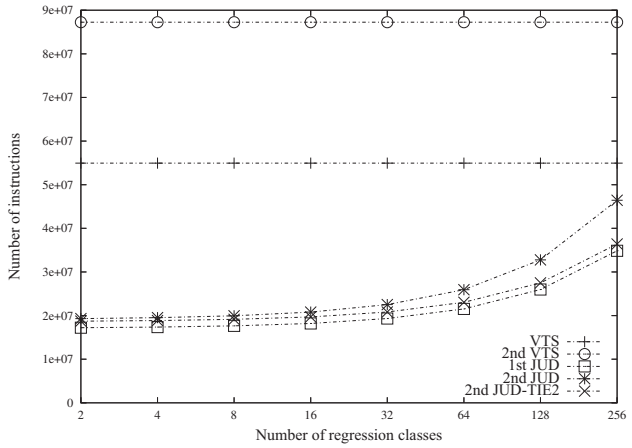

Fig. 2. Computational cost with different number of regression classes

are shown in figure 2 where the number of CPU instructions is measured over one utterance. The JUD-based methods have very limited number of transforms compared to VTS and therefore far less computational cost. Compared to the original JUD, the 2nd-order approximation on JUD does introduce extra computational cost but it is still much faster than first and second-order VTS. Such an extra computational cost can be largely reduced by using the simplified 2nd-order approximation. Given the 6% WER improvement with 32 regression classes on set A, JUD with simplified 2nd-order approximation is about 60% faster than the first-order VTS.

5. CONCLUSIONS

This paper investigates two popular model-based noise robustness methods, namely VTS and JUD. It is observed that JUD is the same as VTS except that JUD is using a rougher expansion point during the Taylor expansion. Due to this reason, JUD cannot beat VTS on the recognition accuracy although it can be much faster. The second-order approximation on JUD is then introduced by virtue of the second-order Taylor expansion, which makes it possible for JUD to beat VTS also on recognition accuracy. To further reduce the overall computational cost, the second-order approximation on JUD is simplified where the derivatives in the Taylor expansion use different number of expansion points in terms of their orders. Compared with VTS, the proposed method achieves up to 6% improvement in recognition accuracy and 60% reduction for the computational cost on the Aurora 2 task, indicating it can be a good replacement for the widely used VTS.

6. REFERENCES

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

[2] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "Hmm adaptation using vector taylor series for noisy speech recogntion," in *Proc.of ICSLP*, Sep. 2000.

[3] 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*, Dec. 2007.

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

[5] H.Xu, L.Rigazio, and D.Kryze, "Vector taylor series based joint uncertainty decoding," in *Proc.of INTERSPEECH*, Sep. 2006, pp. 1125 – 1129.

[6] H. Liao, *Uncertainty decoding for noise robust speech recognition*, Ph.D. thesis, Cambridge University, 2007.

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

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

[9] V.Stouten, H.Van hamme, and P.Wambacq, "Effect of phase-sensitive environment model and higher order vts on noisy speech feature enhancement," in *Proc. of ICASSP 2005*, Philadelphia,USA, March 2005.

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