INTERSPEECH
2009 BRIGHTON

Comparison of Estimation Techniques in Joint Uncertainty Decoding 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

10.21437/Interspeech.2009-369

Abstract

Model-based joint uncertainty decoding (JUD) has recently achieved promising results by integrating the front-end uncertainty into the back-end decoding by estimating JUD transforms in a mathematically consistent framework. There are different ways of estimating the JUD transforms resulting in different JUD methods. This paper gives an overview of the estimation techniques existing in the literature including data-driven parallel model combination, Taylor series based approximation and the recently proposed second order approximation. Application of a new technique based on the unscented transformation is also proposed for the JUD framework. The different techniques have been compared in terms of both recognition accuracy and computational cost on a database recorded in a real car environment. Experimental results indicate the unscented transformation is one of the best options for estimating JUD transforms as it maintains a good balance between accuracy and efficiency.

Index Terms: noise robustness, VTS, joint uncertainty decoding, unscented transformation

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 makes it difficult to deploy ASR technology in real applications.

One approach to tackle this problem is to adapt the previously trained clean speech hidden Markov models (HMM) to the encountered noisy environment. Vector Taylor series (VTS) [1] [2] is a popular technique which applies a linear approximation to 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 as the Taylor series expansion must 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. A key part of JUD is the estimation of JUD transforms based on the probability density function (PDF) of clean speech and the estimated noise. One way to achieve this is by virtue of numerical methods e.g. data-drive parallel model combination (DPMC) [5], which is very costly to compute as a large number of random samples need to be generated. Alternatively, a Taylor series expansion based JUD was introduced in [6] and turns out to be much faster than DPMC. This technique was further compared with the standard first-order VTS in [7] and the two methods were shown to be equivalent except that JUD uses fewer and rougher expansion points. Following this observation, a second order Taylor series expansion was applied to the model-based JUD and seen to give a better performance in comparison to the VTS based JUD.

Recently, unscented transformations (UT) [8] have become a popular method for ASR. This method has the same idea as DPMC but works in a more efficient way by deterministically choosing only a very few important samples. In [9], UT have been adopted for HMM compensation in a way similar to the standard model based VTS i.e. applying UT to each HMM mixture. It shows a recognition performance comparable to DPMC and is much faster. In this paper, the application of UT to the JUD framework for the estimation of JUD transforms is proposed. Compared to [9], this is more advantageous as in this case the UT only needs be conducted for each regression class and therefore the extra computational cost introduced can be minimised.

This paper gives an overview of all the above techniques in JUD and compares them through experiments performed on a database recorded in a real car environment. To our best knowledge, almost all the results reported so far for UT in the literature were based on using artificially corrupted speech data and this is the first paper evaluating this technique on real noisy data. The experiments reported show that UT is one of the best options for estimating the JUD transforms as it gives superior performance both for recognition accuracy and computation cost.

The remainder of this paper is as follows: section 2 gives an overview of model based JUD and all the techniques employed in the estimation of JUD transforms; section 3 compares these techniques through recognition experiments; conclusions are drawn in section 4.

2. Overview of model-based joint uncertainty decoding

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

$$p(x|S) = \sum_{m \in S} c_m p(x|m) = \sum_{m \in S} c_m N(x; \mu_x^m, \Sigma_x^m), \quad (1)$$

where x is the clean speech feature, S is the HMM state, and $N(x; \mu_x^m, \Sigma_x^m)$ is the Gaussian PDF for the mixture m with mean μ_x^m, covariance matrix Σ_x^m and mixture weight c_m.

When noise exists in the input speech, the clean speech feature x is not observable any more. Instead, JUD calculates the output probability of noisy speech feature y of the mixture m as follows:

$$p(y|m) = \int p(y|x, m)p(x|m)dx \tag{2}$$

Depending on how the conditional probability $p(y|x, m)$ is modelled, there exists two JUD methods i.e. feature based and model based. Feature based JUD trains a front-end GMM over clean training data and assumes $p(y|x, m)$ Gaussian for each GMM mixture. Model based JUD uses regression classes [10] to group HMM mixtures and applies the Gaussian assumption to each regression class. In [11], these two methods were thoroughly compared and model based JUD showed much better performance than feature based JUD.

When model based JUD is the case, each HMM mixture m is often assigned to a fixed regression class r_m. Therefore, Eq.(2) becomes

$$\begin{aligned} p(y|m) &= \int p(y|x, r_m)p(x|m)dx \\ &= \int N(y; \mu_{y|x}^{r_m}, \Sigma_{y|x}^{r_m})N(x; \mu_x^m, \Sigma_x^m)dx \\ &= |A_{r_m}|N(A_{r_m}y + b_{r_m}; \mu_x^m, \Sigma_x^m + \Sigma_b^{r_m}) \end{aligned} \tag{3}$$

where

$$\begin{aligned} A_{r_m} &= \Sigma_x^{r_m}(\Sigma_{yx}^{r_m})^{-1}, \\ b_{r_m} &= \mu_x^{r_m} - A_{r_m}\mu_y^{r_m} \\ \Sigma_b^{r_m} &= A_{r_m}\Sigma_y^{r_m}{A_{r_m}}^T - \Sigma_x^{r_m} \end{aligned} \tag{4}$$

In Eq.(4), the clean speech mean $\mu_x^{r_m}$ and covariance $\Sigma_x^{r_m}$ for each regression class can be easily obtained from clean training data. The computation of other parameters, i.e. the mean $\mu_y^{r_m}$ and covariance $\Sigma_y^{r_m}$ for noisy speech and the cross-covariance matrix $\Sigma_{yx}^{r_m}$ depends on the noisy speech. Given noise estimation, their computation has to employ certain estimation technique e.g DPMC, Taylor expansion or UT resulting in different JUD algorithms.

3. JUD with different estimation techniques

3.1. JUD with DPMC

DPMC [12] is a Monte Carlo method. Assuming additive noise feature n Gaussian distributed with the mean μ_n and variance Σ_n, it generates a series of noise feature samples $(n_1, n_2, ...n_s)$ and clean speech feature samples $(x_1, x_2, ...x_s)$ based on their individual distribution $N(n; \mu_n, \Sigma_n)$ and $N(x; \mu_x^{r_m}, \Sigma_x^{r_m})$. Then the corresponding noisy speech features $(y_1, y_2, ..., y_s)$ are obtained by virtue of the popular noise corruption formula:

$$y = x + h + g(x, n, h) = x + h + C\ln(1 + e^{C^{-1}(n-x-h)}) \tag{5}$$

where C denotes the discrete cosine transformation matrix and h the static features for convolutional noise. Finally, $\mu_y^{r_m}$, $\Sigma_y^{r_m}$ and $\Sigma_{yx}^{r_m}$ are calculated based on the y samples:

$$\begin{aligned} \mu_y^{r_m} &= \frac{1}{s}\sum_{i=1}^{s} y_i \\ \Sigma_y^{r_m} &= \frac{1}{s-1}\sum_{i=1}^{s}(y_i - \mu_y^{r_m})(y_i - \mu_y^{r_m})^T \\ \Sigma_{yx}^{r_m} &= \frac{1}{s-1}\sum_{i=1}^{s}(y_i - \mu_y^{r_m})(x_i - \mu_x^{r_m})^T \end{aligned} \tag{6}$$

In [11], DPMC proved to be very powerful when being applied for JUD. However, this is at the expense of a very high computational cost as the number of samples has to be fairly big in order to have a reasonable estimation in Eq.(6).

3.2. JUD with Taylor series based approximations

3.2.1. JUD with first order VTS

In our previous work in [6], we introduced VTS based JUD where the first order Taylor expansion is adopted to linearise Eq.(5) and a closed form solution for the calculation of $\mu_y^{r_m}$, $\Sigma_y^{r_m}$ and $\Sigma_{yx}^{r_m}$ is obtained. Although it was originally proposed for feature based JUD, it can be applied to model based JUD in a similar way. The calculation for $\mu_y^{r_m}$, $\Sigma_y^{r_m}$ and $\Sigma_{yx}^{r_m}$ becomes:

$$\begin{aligned} \mu_y^{r_m} &= \mu_x^{r_m} + h + g(\mu_x^{r_m}, \mu_n, h) \\ \Sigma_y^{r_m} &= W_{r_m}\Sigma_x^{r_m}W_{r_m}^T \\ \Sigma_{yx}^{r_m} &= W_{r_m}\Sigma_x^{r_m} \\ W_{r_m} &= I + \frac{\partial}{\partial x}g(\mu_x^{r_m}, \mu_n, h) \end{aligned} \tag{7}$$

where I is the identity matrix.

Since there is no need to generate a number of samples, there is no doubt that the VTS based JUD in Eq.(7) is more efficient than the DPMC based method. However, it is well known that the Taylor expansion inevitably brings approximation errors and therefore a degradation in recognition accuracy can be observed [11].

3.2.2. JUD with the second order approximation

In [7], JUD with first order VTS was thoroughly analysed and theoretically proven to be the same as the classical model-based first order VTS except that the JUD method uses fewer and rougher expansion points. As a consequence, the VTS based JUD is more efficient but can not beat first order VTS on recognition performance. To overcome this limitation, a new method was introduced by keeping using the same set of expansion points as in the VTS based JUD and embedding the second-order Taylor expansion into the compensation of each HMM mixture. Unlike other methods, this new technique needs to employ a slightly different formula for the HMM compensation:

$$p(y|m) = |A_{r_m}|N(A_{r_m}y + b_{r_m}; \mu_x^m + \Lambda_b^m, \Sigma_x^m + \Sigma_b^{r_m}) \tag{8}$$

where Λ_b^m is a vector obtained per HMM mixture and all the other JUD matrices are computed the same way as in Eq.(7) and (4). According to [7], the above formula only introduces a slight increase in computational cost and the recognition accuracy can exceed the first-order VTS when the number of regression classes increases to a certain level.

3.3. JUD with unscented transformation

UT shares a similar idea to DPMC. However, unlike DPMC which randomly generates a huge number of samples, UT tries to control the number of samples to a minimum level by deterministically adopting a limited number of points in the PDF and assigning certain weights to them. These points, so called sigma points, can be selected in different ways as described in [8].

In [9], UT has been successfully deployed in ASR to compensate HMM models mixture by mixture. Its recognition performance proved to be better than the classical first-order and second-order VTS however at the expense of increasing the computational load, which makes it impractical for small footprint systems. This drawback can be largely mitigated in the JUD framework as UT only needs to be performed per regression class for the calculation of μ_y^{rm}, Σ_y^{rm} and Σ_{yx}^{rm} and the overall increase in the computational cost is expected to be limited.

Similar to [13], this paper selects the sigma points $\{z_i\}_{i=0}^{p}$ for JUD as follows:

$$z_0 = \begin{pmatrix} \mu_x^{rm} \\ \mu_n \end{pmatrix}, \tag{9}$$

$$z_i = z_0 + \left(\sqrt{\frac{N_z}{1-w_0}\Sigma_z}\right)_i \quad (i = 1....N_z), \tag{10}$$

$$z_i = z_0 - \left(\sqrt{\frac{N_z}{1-w_0}\Sigma_z}\right)_{i-N_z} \quad (i = N_z + 1....2N_z) \tag{11}$$

where N_z denotes the dimension of vector z_0, $(M)_i$ means the ith column of matrix M and $p = 2N_z$. The total number of sigma points is $2N_z + 1$.

Given all the sigma points, noisy speech samples $(y_0, y_1, ..., y_p)$ are computed by Eq.(5). Then the mean and variances involved in each regression class for JUD compensation can be acquired by

$$\begin{aligned} \mu_y^{rm} &= \sum_{i=0}^{p} w_i y_i \\ \Sigma_y^{rm} &= \sum_{i=0}^{p} w_i (y_i - \mu_y^{rm})(y_i - \mu_y^{rm})^T \\ \Sigma_{yx}^{rm} &= \sum_{i=0}^{p} w_i (y_i - \mu_y^{rm})(x_i - \mu_x^{rm})^T \end{aligned} \tag{12}$$

where the weights are defined as

$$w_0 = 1 - N_z/3, w_i = (1 - w_0)/(2N_z)$$

4. Experiments

In this section, we compare the performance of different JUD methods. Experiments were conducted on the Toshiba in-car database which was recorded in cars under two driving conditions - engine-on (ENON) and highway (HW). The ENON condition contains 4401 utterances and has an average SNR 35dB, whereas the HW condition contains 4582 sentences with SNR around 18dB. For each noise condition, there are a mixture of small and medium sized tasks including connected digits, command and control and city names. To ease the discussion, all the figures in the following experiments are averaged over all the tasks in each noise condition.

Method	ENON	HW	Average
MTR baseline	6.27	6.76	**6.52**
CM baseline	4.12	68.35	**36.23**
JUD-DPMC(10^4)	2.37	5.59	**3.98**
JUD-VTS	2.90	6.41	**4.66**
JUD-2nd order approx.	1.98	6.37	**4.17**
JUD-UT	2.40	5.69	**4.04**

Table 1: WER (%) averaged over each noise type for different methods.

The front-end employed in this paper is a 13-dimensional MFCC including the zeroth coefficient with their delta and delta-delta components. A triphone HMM with 650 states was trained on a mixed multi-condition set including 312 hours of data consisting of Wall Street Journal, TIDIGITS, TIMIT and internally collected noisy training data. There were 12 mixtures for each speech state in the HMM and 24 mixtures for each silence state, giving the overall number of mixtures in the HMM around 8000. A standard multi-condition training (MTR) HMM was first trained and then refined by joint adaptive training [11]. The final HMM used for JUD compensation and recognition is a canonical model (CM) which is treated as noise free. 16 regression classes were used for JUD for all experiments.

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

1. The initial parameters μ_n, Σ_n and μ_h for noise PDF were estimated from the first and last 20 frames in each utterance.
2. VTS then adapted the HMM to generate an initial recognition hypothesis.
3. An VTS expectation-maximisation based noise estimation process [11] was adopted to refine the noise parameters based on the initial hypothesis.
4. The refined noise parameters were finally fed into JUD to compensate the HMM and obtain the final recognition results. In this paper, only the estimation method is varied, i.e. DPMC, VTS, second order approximation or UT, for the computation of the static part of the JUD matrices. The first-order VTS based method is used for all delta and delta-delta parts.

Table 1 compares the results for JUD with the different estimation techniques discussed in this paper. Results with the MTR HMM and the CM HMM without any compensation are listed here as the baseline.

It is observed that the CM baseline is very poor on the HW condition compared to the MTR baseline but better on the ENON condition. When JUD is in place for compensation, a large gain on the CM HMM can be achieved. Generally speaking, JUD with DPMC gives the best performance on average among all the techniques. However it should be noted the number of samples in DPMC for each regression class plays an important role. In table 2, different values for the number of samples used in DPMC are tested, and 10^4 turns out to be the minimum number of samples required for the DPMC based JUD to achieve a relatively decent performance. This is particularly important for the HW condition.

When VTS is engaged, the result for JUD as shown in table 1 becomes much worse than the DPMC based method. This is reasonable as the Taylor expansion introduces approximation errors.

Method	#Samples per class	ENON	HW	Average
MTR baseline	-	6.27	6.76	**6.52**
JUD-DPMC	10^3	2.42	12.38	**7.40**
JUD-DPMC	$5 * 10^3$	2.37	10.17	**6.27**
JUD-DPMC	10^4	2.37	5.59	**3.98**
JUD-DPMC	10^5	2.39	5.61	**4.00**

Table 2: WER (%) averaged over each noise type for baselines and JUD-DPMC with different number of samples.

JUD +	DPMC (10^4)	VTS	UT	2^{nd} order approximation
# of times using Eq.(5)	$1.6 * 10^5$	16	848	32
MIPS	1411	577	587	580

Table 3: Comparison of computational costs.

The second order approximation helps to boost the JUD performance largely on the ENON condition which is even better than the DPMC. However its gain on the HW condition is limited indicating the second order Taylor expansion is still not sufficient to reduce the Taylor approximation errors to a satisfactory level in noisier conditions. Finally, UT seems to be a very powerful method when working together with JUD. Its performance is almost as good as DPMC for both conditions.

The computational costs of different methods are given in table 3. Two types of measurements are adopted. The first is the number of times of using Eq.(5) when adapting the HMM once with each method. As Eq.(5) is believed to be the major cost for all the JUD methods, this number indicates how costly the estimation part is in the JUD. The second measurement is the CPU cost required for the final step (step 4) of the two-pass decoding which was measured in MIPS (million of instructions per second) over 40 utterances with a total length of roughly 107 seconds. Since this step includes the computation of the JUD matrix and HMM adaptation as well as recognition, it provides an idea how different JUD estimation methods impact on the overall computational cost during decoding. It can be seen that the computational cost of DPMC is much higher than the other three techniques for both measurements. This makes it difficult to deploy in real applications. The two VTS based JUD methods are very cheap. UT has a higher cost than VTS, but their overall CPU costs during decoding are comparable. This is partially because UT is only applied for each regression class and makes the overall increase of computational cost trivial. Considering its superior recognition accuracy as shown in table 1, UT based JUD seems a good option for the balance between high recognition accuracy and low computational cost.

5. Conclusions and future work

This paper gives an overview of different techniques for estimating JUD transforms. These include methods previously presented in the literature e.g. data-driven PMC (DPMC), first order VTS and second order Taylor series approximation as well as a new unscented transformation (UT) based technique. These techniques were compared on a noisy speech database recorded in real cars under engine on and highway driving conditions. In terms of a balance between recognition accuracy and computational cost, experiments showed that overall the UT is superior to other techniques when being applied to JUD.

For less noisy conditions, the second order approximation was found to yield the lowest recognition accuracy, and has a lower computational cost than UT based JUD. Combination of these two noise estimation techniques will be investigated in the future and is expected to further boost the JUD performance.

6. Acknowledgements

The authors would like to thank Yusuke Shinohara in Corporate Research and Development Center, Toshiba Corporation, Japan, for helpful discussions.

7. 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. Liao and M.J.F. Gales, “Uncertainty decoding for noise robust speech recognition,” Tech. Rep., Cambridge University, 2004.

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

[7] H.Xu and K.K.Chin, “Joint uncertainty decoding with the second order approximation for noise robust speech recognition,” in *Proc. of ICASSP*, 2009.

[8] S.J.Julier and J.K.Uhlmann, “Unscented filtering and nonlinear estimation,” *Proc. of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[9] Y.Hu and Q.Huo, “An hmm compensation approach using unscented transformation for noisy speech recognition,” in *Proc. of ISCSLP*, 2006, pp. 346–357.

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

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

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

[13] Y.Shinohara and M.Akamine, “Bayesian feature enhancement using a mixture of unscented transformations for uncertainty decoding of noisy speech,” in *Proc. of ICASSP*, 2009.