Incremental Adaptation with VTS and Joint Adaptively Trained Systems

F. Flego and M.J.F. Gales

Cambridge University Engineering Department, United Kingdom
{ff257, mjfg}@eng.cam.ac.uk

Abstract

Recently adaptive training schemes using model based compensation approaches such as VTS and JUD have been proposed. Adaptive training allows the use of multi-environment training data whilst training a neutral, “clean”, acoustic model to be trained. This paper describes and assesses the advantages of using incremental, rather than batch, mode adaptation with these adaptively trained systems. Incremental adaptation reduces the latency during recognition, and has the possibility of reducing the error rate for slowly varying noise. The work is evaluated on a large scale multi-environment training configuration targeted at in-car speech recognition. Results on in-car collected test data indicate that incremental adaptation is an attractive option when using these adaptively trained systems.

Index Terms: adaptive training, incremental adaptation, noise compensation

1. Introduction

Model-based noise compensation schemes such as Parallel Model Combination (PMC) [1] and Vector Taylor Series (VTS) compensation [2] are powerful approaches to adapting HMM-based speech recognition systems to particular acoustic environments. For these model-based compensation techniques a number of conditions must be satisfied for them to perform well. The impact of the noise on the clean speech must be described by a mismatch function. In addition noise and “clean”-speech model parameters must be available. For parameterisations such as MFCCs mismatch functions can be specified, including the dynamic features [3]. In general the parameters of the noise model are not known and thus must be estimated on the test data using, for example, maximum likelihood estimation [4, 5]. If the speech training data is recorded in high Signal-to-Noise Ratio (SNR) conditions then the “clean” speech models can also be easily obtained. This is the situation described in a range of papers, for example [5, 6].

There are however a couple of problems with this standard scenario. First high SNR condition clean speech training data is not always available. For most noise-robust speech recognition applications the data recorded in the target domain will have a wide-range, and possibly high levels, of background noise. To address this problem adaptive training has been proposed either based on Joint Uncertainty Decoding (JUD) [4] or VTS compensation [7]. The basic concept and mode of training for these approaches is similar to Speaker Adaptive Training (SAT) schemes, e.g. [8]. Rather than treating all the training data as a single block, it is partitioned up into homogeneous blocks. Given a canonical model, transforms for each training block are estimated. Then the canonical model updated given the set of training data condition transforms. In contrast to schemes like SAT the update of the canonical model is made more complicated by the nature of the VTS or JUD transforms. Both the VTS (VAT) and JUD (JAT) adaptive training schemes are related to each other as when the number of regression classes used with JUD is the same as the number of recognition components the two schemes become the same. The approaches differ in the form used to train the canonical model, in [4] a second-order optimisation scheme is used, whereas in [7] an EM-based approach is adopted. To date, though the use of JAT was examined on a broadcast news transcription task, these adaptive training schemes have not been examined on large training datasets with a wide-range of noise conditions.

The second problem is the latency introduced by the need to estimate the noise model for each utterance. The standard approach to doing this [4, 5] is to perform multiple recognition and noise estimation runs over the training data, thus operating in a batch adaptation mode. The larger the number of recognition/noise estimation iterations the greater the latency. In [9] this latency issue was addressed by using an incremental adaptation mode, where the, possibly smoothed, noise estimates from the previous utterance were used to decode the current utterance. This approach was evaluated using “clean” acoustic models.

This paper examines the application of incremental adaptation to adaptively trained systems with large amounts of acoustic training data. This combination of approaches, as well as addressing both limitations with the standard model-based compensation approach, can potentially handle another problem with adaptively trained systems. These systems cannot be directly used for decoding as the canonical model parameters are trained under the assumption that a test set transform is available. Thus a multi-style trained system is often used to get an initial hypothesis for each utterance. When using incremental adaptation this is not an issue as the transform from the previous utterance can be used. A further problem that can be encountered with JAT or VAT systems in low SNR conditions on more complex tasks is that the initial hypothesis from a multi-style trained system may be poor. This means that a large number of recognition/noise estimation iterations may be required to obtain good performance. With incremental adaptation this will not be as large a problem as, provided the noise does not change too much from iteration to iteration, the transform will be gradually refined as the number of utterances increases.

2. Model-based compensation

In this work VTS and Joint Uncertainty Decoding (JUD) adaptation approaches are used. These methods are described in detail in [2, 10], and only a brief description is given here. VTS and Joint are model based noise compensation approaches. The require a mismatch function, describes the relationship between the clean and corrupted speech parameters, to be specified. The HMM feature vector at time t normally contains static, delta and

This work was partly funded by Toshiba Research Europe Ltd (TREL). The authors would also like to thank TREL for providing data.
delta-delta coefficients, \( y_i^T = [y_i^T, y_i^{ΔT}, y_i^{Δ2T}] \). For MFCC features the following form of static mismatch function in the mel-cepstral domain can be used

\[
y_i^T = x_i^T + h + C \log \left( 1 + \exp(C^{-1}(n_i - x_i^T)) \right)
\]

where \( C \) is the DCT matrix, \( x_i^T \) and \( n_i \) are the clean speech, convolutional noise and noise vectors respectively. Given the noise parameters \( M = \{ \mu_s, \Sigma_s, \mu_h \} \), it is possible to use Equation (1) to adapt the model parameters.

For VTS compensation, considering Equation (1), the static mean, \( \mu_s^T \), and covariance matrix, \( \Sigma_s^T \), of the corrupted speech distribution are given by

\[
\mu_s^T = \mu_s^T + \mu_h + f(\mu_s^T - \mu_s - \mu_h)
\]

\[
\Sigma_s^T = \text{diag}(\{ \Sigma_s^T \Delta + (I - J) \Sigma_s^T \Delta (I - J)^T \})
\]

where the matrix \( J \) is the partial derivative, \( \partial y_i^T / \partial x^T \), evaluated at \( \mu_h = \mu_s - \mu_s^T = \mu_h^T \). To obtain the expression for the compensated dynamic parameters (\( \Delta, \Delta^2 \)) the continuous time approximation [3] is used providing

\[
\mu_s^\Delta = J \mu_s^\Delta;
\]

\[
\Sigma_s^\Delta = \text{diag}(\{ J \Sigma_s^\Delta + (I - J) \Sigma_s^\Delta (I - J)^T \})
\]

where \( \mu_s^\Delta \) and \( \Sigma_s^\Delta \) have similar form. For each component \( m \) the likelihood of the corrupted speech is obtained as

\[
p(y_i^T | m) = N(y_i, \mu_s^\Delta (m), \Sigma_s^\Delta (m))
\]

Though VTS has been shown to yield large reductions in word error rate, the scheme is computationally expensive as each system component has to be compensated individually.

To reduce this computational load, JUD has been proposed [10]. JUD estimates compensation parameters at the base-class level. The joint Gaussian distribution of the corrupted speech, \( y_i \), and the clean speech, \( x_i \), is estimated for each regression class \( r \). This can then be used to compute the likelihood of each component \( m \) belonging to regression class \( r \) as

\[
p(y_i | m) = |A^{(r)}| N(A^{(r)} y_i + b^{(r)}, \mu_s^{(m)}, \Sigma_s^{(m)} + \Sigma_b^{(r)})
\]

The computational advantage of JUD is that the noise transform \( T^{(r)} = \{ A^{(r)}, b^{(r)}, \Sigma_b^{(r)} \} \) is obtained by applying noise compensation at the regression class, rather than component level. These transforms are then efficiently applied to the system components as only a variance bias needs to be added.

3. JUD and VTS Adaptive Training

Adaptive training using JUD or VTS is carried out in an EM framework. Two stages are used: first a new set of transforms \( T \) is estimated given the current canonical model \( \mathcal{M} \). Then new canonical model parameters \( \mathcal{M} \) are estimated given the current transforms \( T \). Multiple iterations interleaving transforms and canonical model estimation are generally applied [8, 10]. The estimation of transform parameters is described in the previous section. In this section the estimation of the canonical model for JAT [4] is given. However, since when the number of regression classes used in the Joint case is set equal to the number of system components, the same formula can be applied to VAT.

The estimation of the canonical model is based on maximising the following objective function:

\[
Q\left( \mathcal{M}, T; \mathcal{M}, T \right) = \sum_{h=1}^{H} \sum_{t=1}^{T_h} \sum_{m=1}^{M} \gamma_i^{(mh)} \times \log \left( \frac{1}{\gamma_i^{(mh)}} \right)
\]

where \( \gamma_i^{(mh)} \) is the posterior probability that the observation \( y_i \) is generated by component \( m \) on heterogeneous training data segmented into \( H \) homogeneous blocks, each of length \( T_h \), for all valid state sequences given the transcription.

This function can be maximised w.r.t \( \mathcal{M} \) using a second-order gradient based optimisation scheme. For component \( m \)

\[
\left[ \begin{array}{c}
\mu_{s_{(m)}}^{(m)} \\
\sigma_{s_{(m)}}^{(m)2}
\end{array} \right] = \left[ \begin{array}{c}
\mu_{s_{(m)}}^{(m)} \\
\sigma_{s_{(m)}}^{(m)2}
\end{array} \right] + \left[ \begin{array}{c}
\left( \frac{\partial^2 Q}{\partial \mu_{s_{(m)}}^{(m)} \partial \mu_{s_{(m)}}^{(m)}} \right) \\
\left( \frac{\partial^2 Q}{\partial \mu_{s_{(m)}}^{(m)} \partial \sigma_{s_{(m)}}^{(m)2}} \right)
\end{array} \right]^{-1} \left[ \begin{array}{c}
\left( \frac{\partial Q}{\partial \mu_{s_{(m)}}^{(m)}} \right) \\
\left( \frac{\partial Q}{\partial \sigma_{s_{(m)}}^{(m)2}} \right)
\end{array} \right]
\]

with \( \mu_{s_{(m)}}^{(m)} = \gamma_{i_{(mh)}}^{(m)} / \sigma_{s_{(m)}}^{(m)2} + \sigma_{b_{(m)}}^{(m)2} \). Similar expressions can be found for the Hessian matrix terms. It is worth noting that each term in the summation is weighted by \( \gamma_{i_{(mh)}}^{(m)} \). For observations with lower SNR the JUD uncertainty bias term \( \sigma_{b_{(m)}}^{(m)2} \) becomes large, decreasing the value of \( \omega_{i_{(mh)}}^{(m)} \). This in turn will de-weight the contribution of such observations to the estimate of the auxiliary function derivatives. This agrees with the intuition behind adaptive training.

4. Incremental Adaptation

One of the problems with adaptively trained systems is that without a transform suitable for test set adaptation they cannot be directly used for recognition. This is not as much of a problem with predictive schemes such as VTS and JUD where robust adaptation can be achieved using a single utterance, compared to SAT [8]. However initial hypotheses to derive the transforms are still required. The standard approach for decoding with an adaptively trained system is as follows

1. For each utterance: \( i = 1 \) to \( N \), \( Y^{(i)} \)
2. Initial transform: for the case of linear transform-based adaptation an identity transform is often used \( T_0^{(i)} = I \).
3. Initial hypothesis: this is usually obtained from a multi-environment/speaker trained system. This yields the initial hypothesis \( H_0^{(i)} \)
4. Initialise the loop-count \( k = 1 \).
5. Transform estimation: given the current hypothesis, \( H_{k-1}^{(i)} \), and transform, \( T_{k-1}^{(i)} \), an ML-estimate of the transform parameters is obtained using an EM-based approach. This stage can be repeated using the same hypothesis, where the transform is repeatedly estimated.
improving the component posteriors associated with the EM estimation. The output from this stage is a new estimate of the transform $T_k^{(i)}$.

6. **Hypothesis generation**: given the transform estimated in the previous stage, $T_k^{(i)}$, the utterance is decoded to yield a new hypothesis $H_k^{(i)}$.

7. if reached max iteration, $atop$, else $k = k + 1$, goto (4).

This form of adaptation will be referred to as batch-mode adaptation in this paper.

The general process described above can be slightly modified when using predictive approaches [4, 5]. The initial transform can be estimated from, for example, the first and last 20 frames of the utterance and then the adaptively trained system used for decoding. However, since the system is adaptively trained, in initial experiments this was found to yield poorer performance compared to unadapted multi-style trained system and so is not used.

An alternative approach proposed in this paper is to use an incremental adaptation framework. This process is described below.

1. **Initial transform**: for the case of linear transform-based adaptation an identity transform is often used $T^{(0)} = I$.

2. **For each utterance**: $i = 1$ to $N$, $Y^{(i)}$

3. **Hypothesis generation**: given the transform estimated in the previous utterance, $T^{(i-1)}$, the utterance is decoded to yield a new hypothesis $H^{(i)}$.

4. **Transform estimation**: given the current hypothesis, $H^{(i)}$, and initial transform, $T^{(0)}$, an ML-estimate of the transform parameters is obtained using an EM-based approach.

5. $i = i + 1$, goto (3).

This form of adaptation will be referred to as incremental-mode adaptation in this paper. In previous work this form of adaptive training was applied to clean models [9]. It was motivated from a need to reduce the latency associated with the standard batch-mode approach where multiple recognition runs, and transform (noise) estimation stages, are required.

This form of incremental adaptation is well suited for adaptively trained systems which make more use of the training/test transforms than multi-environment systems. The canonical models are trained given a transform, and that transform is estimated during training on the correct hypothesis. Thus adaptively trained systems are typically more sensitive to both the hypothesis used during test-set adaptation and the number of EM iterations used in step (5) of the batch-mode adaptation. The incremental mode described above addresses these problems to some extent. Provided that the noise does not rapidly change from utterance to utterance, the transform from the previous utterance should be a good estimate for the current utterance. This should improve both the hypothesis generated and the estimate of the transform, as the initial transform will be better. Thus provided the noise only changes slowly incremental adaptation can become the equivalent of many iterations of batch adaptation. Another advantage of incremental mode adaptation is that decoding with the adaptively trained system is possible, provided that the initial transform is good enough (though an initial transform for the first utterance of a speaker is still required).

Depending on the nature of the noise it is also possible to smooth the noise estimates over multiple utterances. This reduces the sensitivity of the transform estimate to errors in the hypothesis. However, this smoothing process can degrade performance if the noise is rapidly changing. When smoothing is applied there are a number of approaches that can be adopted. One efficient approach is to smooth previous transform estimates with the current estimate. The approach adopted here, which is slower but more accurate is to smooth the statistics that the noise transform is estimated on. Thus if the statistics for estimate the transform the form for utterance $i$ is $\mathbf{O}^{(i)}$ then the statistics used to estimate the transform for utterance $i + 1$ is

$$\mathbf{O}^{(i+1)} = \alpha \mathbf{O}^{(i)} + \mathbf{O}^{(i)}$$

where the statistics collected from utterance $i + 1$ are

$$\mathbf{O}^{(i+1)} = \left\{ \sum \gamma^{(m)}_i \sum \gamma^{(m)}_{i+1} y_i \sum \gamma^{(m)}_{i+1} y_{i+1} \right\}$$

$\alpha$ can be tuned depending on the nature of the noise. For all experiments it was set to 0.6.

5. **Experimental Results**

The performance of the proposed scheme was evaluated on an in-car recognition task provided by TREL. The training data for this task consists of 486 hours of acoustic training data. This comprises both artificially corrupted clean speech data with car noise added at a range SNR levels and in-car collected training data. The feature used were 12 MFCCs appended with the zero cepstrum, and delta and delta-delta coefficients. The total number of decision tree clustered states was about 650 with 12 Gaussian components per state and diagonal covariance matrices. Cross-word triphones models were used with three emitting states per HMM. This system is more compact than the usual form of system built on this size of data, but is felt to be more realistic for an embedded application. For JUD and JAT a regression class tree with 64 classes was used, which amounts to approximately a hundredth the number of components in the system. The training data was used to build both multi-style and adaptively trained systems.

The test data used consists of four subtasks recorded either with the engine-on (ENON) or driving along a highway (HWAY). The SNR-levels for these two configurations were approximately 35dB and 18dB respectively. The four small/medium sized tasks were: unknown-length digit sequence recognition phone numbers (PH), four digits recognition (4D), command & control (CC) and city names (CN). The first two contain digit sequences, the third contains in car radio or navigator command and control sequences with a vocabulary of 119, while the fourth consists of utterances with a single city name each with a vocabulary of 544. All test sets were recorded in a car with a microphone mounted on the rear-view mirror. Though these are not, for example, spontaneous dialog tasks, it is in-car recorded data which allows an initial investigation of the proposed approach. The test sets comprise about 30 speakers, each uttering 30 sentences (60 in the CC task). The approximate size of the four test sets in hours is 1.5, 0.75, 2.5 and 0.85 respectively.

For the batch experiments the initial hypothesis was obtained from the multi-style trained system and the initial noise transform obtained from the first and last 20 frames of each utterance setting $\mu_s = 0$. The hypothesis was updated at each iteration. Initial transforms for each first utterance were obtained in the same way for the incremental-mode.

Initially a contrast of the performance of using multi-style training or VAT in batch-mode adaptation was performed with
the PH task. The results are shown in Table 1. Both systems used the same initial hypothesis from the unadapted multi-style system (iteration 0). For the high SNR ENON condition the use of VTS degraded the performance of the multi-style trained system. This is because the mismatch function used assumes a clean acoustic model, rather than the multi-style one provided. For the low SNR condition HWAY the use of VTS gave gains for the multi-style trained system. However the adaptively trained system out-performed the multi-style system for both ENON and HWAY, with VTS always yielding gains, illustrating that the VAT neutral model is more appropriate for VTS compensation. The rest of the experiments are based on adaptively trained systems.

Table 1: WER(%): multiple batch VTS iterations, comparison between multi-style and adaptively trained system, PH test set.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>ENON</th>
<th>HWAY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>multi</td>
<td>adapt</td>
</tr>
<tr>
<td>0</td>
<td>1.1</td>
<td>4.5</td>
</tr>
<tr>
<td>1</td>
<td>1.5</td>
<td>0.5</td>
</tr>
<tr>
<td>2</td>
<td>1.4</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 2: WER(%): multiple batch VTS iterations on VAT system, all test sets HWAY condition.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>HWAY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PH</td>
</tr>
<tr>
<td>0</td>
<td>4.5</td>
</tr>
<tr>
<td>1</td>
<td>1.6</td>
</tr>
<tr>
<td>2</td>
<td>1.5</td>
</tr>
</tbody>
</table>

The second experiment examined the sensitivity to the initialisation. Table 2 shows the performance for multiple VTS iterations with the VAT trained system using all the test sets in the HWAY noise condition. For the digit string recognition tasks (PH and 4D) and the CC task the VTS adaptation worked well. However for the CN task, which has the highest initial error rate and relatively short utterances, the overall performance was poor. If the number of iterations was increased the performance on the CN task carried on improving, but in practice using these large numbers of recognition/noise estimation iterations is impractical.

Table 3: WER(%): Comparison between 2-iterations batch VTS on VAT system and incremental adaptation on VAT and JAT systems; ETSI advanced Front-End results on multi-style system

<table>
<thead>
<tr>
<th>System</th>
<th>ENON</th>
<th>HWAY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PH</td>
<td>4D</td>
</tr>
<tr>
<td>VAT-batch</td>
<td>0.5</td>
<td>0.2</td>
</tr>
<tr>
<td>VAT-inc</td>
<td>0.6</td>
<td>0.3</td>
</tr>
<tr>
<td>JAT-inc</td>
<td>1.2</td>
<td>0.7</td>
</tr>
<tr>
<td>ETSI-adv</td>
<td>1.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Incremental-mode adaptation was then compared to batch-mode. In Table 3 the batch-mode VTS adaptation on a VAT system is compared with incremental-mode adaptation on both a VAT and JAT systems. The incremental-mode VAT achieves about the same WER on ENON noise as the batch-mode VAT system. For the three tasks on which the batch-mode VAT system performed well in the HWAY condition, a degradation in performance is observed in going to incremental-mode. This is expected as the the noise estimate from the previous utterance is being used. However on the CN task in the HWAY condition incremental-mode adaptation shows large gains over the batch-mode performance. Incremental mode has addressed the sensitivity to the initialisation for this task. In addition JAT, with JUD incremental adaptation was run. This is fast compared to VTS as both noise estimation and compensation is performed primarily at the regression class level. As expected JAT doesn’t perform as well as the incremental-mode VAT, but is on average better than the batch-mode VAT. In addition a system built using the same training data with the ETSI advanced front-end [11] was evaluated on this task. For the majority of the test-sets incremental VAT out-performed the ETSI advanced front-end and had on average about 20% relative lower WER.

6. Conclusions

This paper has examined this use of incremental-mode adaptation with VTS and JUD adaptively trained systems. The use of incremental-mode adaptation addresses some of the problems with the standard batch-mode adaptation, in particular the latency from estimating transforms using the current utterance. Furthermore the use of incremental-mode adaptation, when the noise is slowly varying, improves the transform estimation as the transform will be improved after each utterance. The performance of the system was evaluated using a relatively large-scale multi-environment training corpus, including both artificially corrupted data and in-car data. In-car collected data from four small/medium size tasks was used to assess the performance. Overall incremental-mode adaptation yielded a lower average error rate than batch-mode adaptation, or using the ETSI advanced front-end. This indicates that on real-noise data the incremental-mode adaptation has good potential benefits for adaptively trained systems.

7. References