ADAPTIVE TRAINING FOR ROBUST ASR

M.J.F. Gales

Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, UK
mjfg@eng.cam.ac.uk

ABSTRACT

Adaptive training is a powerful training technique for building speech recognition systems on non-homogeneous data. The aim is to remove unwanted variability, such as changes in speaker, channel or acoustic environment, from desired changes, the acoustic differences between words. During training two sets of models are generated; a canonical model set for the desired "true" variability of the speech data, and a set of transforms to represent the unwanted variability. The canonical model set trained in this fashion should be more "amenable" to being adapted to a particular target condition and more "compact". During recognition a transform to the target domain is trained. This target specific transform is then used with the canonical model set in the recognition process. The underlying theory and assumptions used in adaptive training are examined in this paper, Furthermore, the use of adaptive training schemes in current state-of-the-art tasks is described, together with a discussion of how such schemes may be used in the future.

1. INTRODUCTION

The traditional approach to achieving robust speech recognition, is to build a speech recognition system on "clean" training data. This model set is then *adapted* to the target acoustic environment, or the corrupted feature vectors *enhanced*. In recent years there has been a trend towards using *found* training data, rather than specially collecting training data, to build speech recognition systems. Found data generally has greater variability than the specially collected data. In addition to the variability resulting from multiple speakers there are typically more changes in acoustic and channel conditions. The usual approach to handle this non-homogeneous training data is to rely on the frontend feature extraction process to remove the unwanted variability, or acoustic factors. A model set is then built on the set of features as if they all came from a single, consistent, source. Current feature extraction techniques do not perfectly remove all the factors, there is variation in the acoustic feature vectors, sometimes dramatic, as the speaker and acoustic conditions change. The acoustic models must account for this additional variability. For this reason systems built in this fashion are commonly known as *multi-style* trained systems. Though good performance has been obtained with multi-style systems, it would be preferable to have a training scheme that is more appropriate for building systems on found data. As the use of found data increases techniques to handle this variability intelligently will become more important. This paper presents adaptive training as one possible scheme to handle the variability.

Adaptive training is a powerful training technique for building speech recognition systems on non-homogeneous data. Though simple adaptive training schemes are used in all state-of-the-art speech recognition systems, the use of more complex adaptive training schemes is less common. This paper examines some of the reasons for this and possible solutions to the problems. Adaptive training was originally proposed to handle speaker differences [1]. However, adaptive training schemes may also be used to train systems on data from multiple acoustic environments. The basic concept of adaptive training is to train one or more transforms to represent each training speaker and acoustic environment. A *canonical* model is then trained, given the set of speaker environment transforms. This canonical model should represent only the inherent variability of the data. As such it should be more compact than the multi-style systems since the multi-style systems must also model the speaker and acoustic variabilities. Furthermore, due to the nature of the training the model set should be more amenable to being transformed to a new speaker, or acoustic, condition than standard multi-style systems. Of course if an appropriate set of feature vectors that are inherently robust to speaker and environment changes is developed then the need for adaptive training is removed. Unfortunately the most popular frontends for speech recognition, MFCCs and PLPs, do not achieve this goal and, to the author's knowledge, such a frontend does not exist.

Adaptive training schemes may be split into three broad classes. These are:

1. **Model independent**: these schemes do not make explicit use of any model information. The two most common forms are cepstral mean normalisation (CMN) and cepstral variance normalisation (CVN). These transforms are directly applied to the features.

2. **Feature transformation**: these transforms also act on the features but are derived, normally in using maximum likelihood estimation (MLE), using the current estimate of the model set. Common versions of these feature transforms are vocal tract length normalisation (VTLN) [2] and constrained MLLR [3].

3. **Model transformation**: the model parameters, means and possibly variances, are transformed. Common schemes are the original speaker adaptive training (SAT) [1] using maximum likelihood linear regression (MLLR) [4], and cluster adaptive training (CAT) [5].

The first class, model-independent adaptive training, will be referred to as a simple adaptive training scheme. The other two as complex adaptive training schemes. This paper examines the advantages and disadvantages of using adaptive training schemes for robust ASR. Possible future directions for their use of such schemes are also described.

2. ADAPTIVE TRAINING

The original motivation for adaptive training was based on integrating the adaptation scheme (MLLR) into the training process [1]. This papers describes adaptive training in a similar fashion, but presented as a form of graphical model. This allows some of the proposed schemes to be simply derived and motivated. Standard HMMs may be viewed as a (simple) dynamic Bayesian network. The addition of adaptive training yields the network of figure 1.

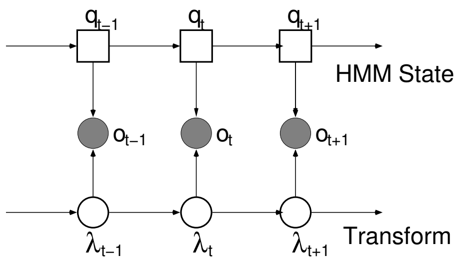

Fig. 1. Dynamic Bayesian network for adaptive training

It is normally assumed that the speaker/acoustic condition is constant over a "block" (usually sentence) of data. Thus over a block of data $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t$. Other constraints, for example that the transform is itself generated by a Markov process, are possible [6], but have not been applied for complex continuous transforms. The likelihood of a sequence of feature vectors in block s, $\mathbf{O}^{(s)} = \mathbf{o}_1^{(s)}, \dots \mathbf{o}_T^{(s)}$, is given by

$$\begin{aligned} p(\mathbf{O}^{(s)}|\mathcal{M}, \tilde{\mathcal{H}}) &= \int p(\mathbf{O}^{(s)}, \boldsymbol{\lambda}|\mathcal{M}, \tilde{\mathcal{H}}) d\boldsymbol{\lambda} \\ &= \int p(\mathbf{O}^{(s)}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \tilde{\mathcal{H}}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda} \end{aligned} \quad (1)$$

where $\mathcal{M} = \{\boldsymbol{\Theta}, \boldsymbol{\phi}\}$ are the sets of model parameters. and $\tilde{\mathcal{H}}$ is the (correct) data transcription. If the canonical model is an HMM then

$$p(\mathbf{O}^{(s)}|\mathcal{M}, \tilde{\mathcal{H}}) = \quad (2)$$
$$\int \sum_{\{\mathbf{Q}^{(\tilde{\mathcal{H}})}\}} \left(\prod_{t=1}^{T} p(\mathbf{o}_t^{(s)}|\boldsymbol{\theta}_{q_t}, \boldsymbol{\lambda}) \right) P(\mathbf{Q}|\boldsymbol{\theta}_d) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda}$$

where $\{\mathbf{Q}^{(\tilde{\mathcal{H}})}\}$ is the set of all valid state sequences through the model of length T for the transcription, $\tilde{\mathcal{H}}$, and q_t is the state at time t along the particular path $\mathbf{Q}$. From the training data we need to extract two distinct sets of model parameters.

1. **Canonical model parameters**: this models the acoustic data given the "unwanted" acoustic factor transform. For the work in this paper a standard HMM is used for this model. Thus the set of model parameters $\boldsymbol{\Theta}$ consists of the state probability density functions for each state q, $\boldsymbol{\theta}_q$, and the state sequence probability (duration model) parameters, $\boldsymbol{\theta}_d$.
2. **Transform distribution**: these represent the variation over the training transforms. The set of parameters to be trained is denoted as $\boldsymbol{\phi}$. The exact form of the prior transform distribution, $p(\boldsymbol{\lambda}|\boldsymbol{\phi})$, is important, particularly when no form of test set adaptation is used.

The model parameters are usually trained MLE. Here

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \left(\sum_{s=1}^{S} \log(p(\mathbf{O}^{(s)}|\mathcal{M}, \tilde{\mathcal{H}})) \right) \quad (3)$$

where S is the number of training blocks. In common with standard HMM training the model parameters may be estimated using expectation-maximisation (EM) [7]. The following auxiliary function is optimised with respect to $\hat{\mathcal{M}}$

$$\mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) = \quad (4)$$
$$\sum_{s=1}^{S} \int p(\boldsymbol{\lambda}|\mathbf{O}^{(s)}, \mathcal{M}, \tilde{\mathcal{H}}) \log \left(p(\boldsymbol{\lambda}, \mathbf{O}^{(s)}|\hat{\mathcal{M}}, \tilde{\mathcal{H}}) \right) d\boldsymbol{\lambda}$$

where $\mathcal{M}$ is the old model set and $\hat{\mathcal{M}}$ is the "new" set of model parameters. Unfortunately it is not possible to directly estimate the model parameters from this expression. Instead it is normally assumed that there is sufficient data for each block s such that

$$p(\boldsymbol{\lambda}|\mathbf{O}^{(s)}, \mathcal{M}, \tilde{\mathcal{H}}) \approx \delta(\boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}^{(s)}) \quad (5)$$

where the ML estimate of the transform parameters is used

$$\hat{\boldsymbol{\lambda}}^{(s)} = \arg\max_{\boldsymbol{\lambda}} \left(p(\mathbf{O}^{(s)}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \tilde{\mathcal{H}}) \right) \quad (6)$$

Now equation 4 may be written as

$$\begin{aligned} \mathcal{Q}(\mathcal{M}, \hat{\mathcal{M}}) &= \sum_{s=1}^{S} \log(p(\hat{\boldsymbol{\lambda}}^{(s)}, \mathbf{O}^{(s)}|\hat{\mathcal{M}}) \\ &= \mathcal{Q}(\mathcal{M}, \hat{\boldsymbol{\Theta}}) + \mathcal{Q}(\mathcal{M}, \hat{\boldsymbol{\phi}}) \end{aligned} \quad (7)$$

where

$$\mathcal{Q}(\mathcal{M}, \hat{\boldsymbol{\Theta}}) = \sum_{s=1}^{S} \log(p(\mathbf{O}^{(s)}|\hat{\boldsymbol{\lambda}}^{(s)}, \hat{\boldsymbol{\Theta}}) \quad (8)$$

and

$$\mathcal{Q}(\mathcal{M}, \hat{\boldsymbol{\phi}}) = \sum_{s=1}^{S} \log(p(\hat{\boldsymbol{\lambda}}^{(s)}|\hat{\boldsymbol{\phi}})) \quad (9)$$

Equation 8 for the case of an HMM canonical model gives the "standard" adaptive training scheme [1]. Equation 9 requires the estimation of a generative model for the transform parameters. The exact form of the optimisation required for each of these equations depends on the transformation being used. In the original formulation the transform parameter distribution was fixed as uniform over the transform parameter space and never updated.

Adaptive training is an iterative training process. There are two distinct stages. First, given the current set of model parameters, the transform parameters are estimated for each block using equation 6. The estimation of the transform parameters for a block may itself be an iterative process based on EM, for example MLLR [4]. Then, given the transform parameters, the model parameters are updated using equation 8 and the process repeated. The estimation of the model parameters is itself an iterative process as the HMM model parameters are usually estimated using EM. If the transform parameter distribution is also to be estimated this would add yet another stage in the iterative training process.

The standard scheme for using adaptively trained schemes during recognition is to estimate the target domain transform based on some supervised adaptation data $\mathbf{O}^{(a)}$ (i.e. the correct transcription of the adaptation data is know). MLE is used to estimate the transform

$$\hat{\boldsymbol{\lambda}}^{(\tilde{\mathcal{H}})} = \arg\max_{\boldsymbol{\lambda}} \left(p(\mathbf{O}^{(a)}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \tilde{\mathcal{H}}) \right) \tag{10}$$

This transform parameter value is then used in the decoding process for subsequent data $\mathbf{O}$. For hypothesis $\mathcal{H}$

$$\begin{aligned} p(\mathbf{O}|\mathcal{M}, \mathbf{O}^{(a)}, \tilde{\mathcal{H}}, \mathcal{H}) &\approx p(\mathbf{O}|\boldsymbol{\Theta}, \hat{\boldsymbol{\lambda}}^{(\tilde{\mathcal{H}})}, \mathcal{H}) \qquad (11) \\ &= \sum_{\{\mathbf{Q}^{(\mathcal{H})}\}} \left(\prod_{t=1}^{T} p(\mathbf{o}_t|\boldsymbol{\theta}_{q_t}, \hat{\boldsymbol{\lambda}}^{(\tilde{\mathcal{H}})}) \right) P(\mathbf{Q}|\boldsymbol{\theta}_d) \end{aligned}$$

As the estimation of the transform parameters is usually based on EM it is important to use multiple iterations of transform estimation. The number of iterations required is dependent on how close the initial estimate of the transform is to "true" transform parameters.

Adaptive training is a very attractive scheme for training speech recognition systems on found data as it can handle non-homogeneous training data. Yet the use of adaptive training schemes, other than the simplest model-independent transforms, is not as widespread as might be expected. This may partly be attributed to the additional complexity in training the model sets. However this is "simply" a software problem. The main reason for the limited use is the use of a point MLE estimate for the test adaptation transform. When only limited adaptation data is available there is little confidence (i.e. large variance) on the estimate of the transform parameters. Simply decoding with the ML estimate may degrade consequent performance. This problem may be expected to be greater for adaptively trained systems, which rely more on "correct" transformations than standard multi-style systems. This problem becomes even worse when unsupervised adaptation is used as described in the next section. Despite these problems good performance gains have been found using complex adaptive training schemes for supervised adaptation [1, 3].

3. UNSUPERVISED ADAPTATION

The section above has assumed that supervised adaptation is being used during recognition. For many tasks there is no suitable, correctly transcribed, adaptation data. In these situations unsupervised adaptation techniques are required. Unsupervised adaptation comes in two basic forms. The first, *incremental* adaptation allows the test data to be decoded only once. An alternative, common scenario, is *transcription* mode adaptation. Here it is possible to recognise the data multiple times and, if desired, use it as adaptation data to recognise itself. This will be referred to as *self adaptation*. This section will describe the issues of using adaptive training in an unsupervised adaptation mode. In particular the use of adaptive training with self adaptation is described. Similar arguments hold for unsupervised incremental adaptation.

If possible during decoding the *correct* set of transform parameters, $\tilde{\boldsymbol{\lambda}}$ would be used. However, as there is no adaptation data available it is not possible to obtain an estimate of this correct transform. The problem is to obtain a set of transform parameters "close" to the true parameters with no explicit adaptation data. The following approximation may be used to obtain an estimate of the transform parameters.

$$\begin{aligned} p(\mathbf{O}|\mathcal{M}) &= \sum_{\mathcal{H}} \int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda} P(\mathcal{H}) \\ &\leq \int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \check{\mathcal{H}}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda} \\ &\leq p(\mathbf{O}|\hat{\boldsymbol{\lambda}}^{(\check{\mathcal{H}})}, \boldsymbol{\Theta}, \check{\mathcal{H}}) \end{aligned} \tag{12}$$

where

$$\check{\mathcal{H}} = \arg\max_{\mathcal{H}} \left(\int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda} P(\mathcal{H}) \right) \tag{13}$$

and $\hat{\boldsymbol{\lambda}}^{(\check{\mathcal{H}})}$ is obtained using MLE and the hypothesis $\check{\mathcal{H}}$. The first approximation assumes that only the MAP estimate of the transcription, rather than the sum over all possible hypothesis, is required. Second it is assumed that the ML estimate of the transform parameters, rather than the MAP estimate or posterior distribution, is sufficient. After each approximation the likelihood of the data, $\mathbf{O}$, is guaranteed to be greater than or equal to the previous likelihood. Using these approximations a set of transform parameters, $\hat{\boldsymbol{\lambda}}^{(\check{\mathcal{H}})}$, are obtained that may be used in decoding. The hope is that the error rate of the hypothesis, $\hat{\mathcal{H}}$, using the transform, $\hat{\boldsymbol{\lambda}}^{(\check{\mathcal{H}})}$, is lower than that for the original hypothesis, $\check{\mathcal{H}}$. It is not necessary to only do a single update of the hypothesis. It is possible to repeat this whole process by generating a new transform, $\hat{\boldsymbol{\lambda}}^{(\hat{\mathcal{H}})}$ with the new hypothesis $\hat{\mathcal{H}}$ and recognising again. This is the basis of schemes such as iterative MLLR [8].

Though the likelihood of the test data is guaranteed to not decrease with the estimated transform, there are a couple of major issues with this form of adaptation, in addition to the problems described for supervised adaptation. The first problem is that for each of the approximations made the system becomes more closely "tuned" to the hypothesised transcription $\check{\mathcal{H}}$. If the complexity of the transform is too high, or the length of the sentence is too short, then there will be a strong tendency for the new hypothesis $\hat{\mathcal{H}}$ to equal the original hypothesis $\check{\mathcal{H}}$, yielding no performance gain. Indeed, this has been used as a stopping criterion for increasing the complexity of the transformation with MLLR [9]. The need to balance the tuning to one particular hypothesis is one of the important issues with self adaptation.

The second major issue with unsupervised "self" adaptation is how to obtain the original hypothesis, $\check{\mathcal{H}}$. The expression given in equation 13 has no simple closed-form solution. Instead some approximation is required. The simplest solution is to use the unadapted model parameters in the initial decoding,

$$\check{\mathcal{H}} = \arg\max_{\mathcal{H}} \left(p(\mathbf{O}|\boldsymbol{\Theta}, \mathcal{H}) P(\mathcal{H}) \right) \tag{14}$$

In some systems, notably the original work by BBN, decoding in this fashion yielded only a small increase in the error rate of the initial hypothesis compared to a multi-style system. However, on other systems it has been found to give poor performance [10]. Alternatively a multi-style model, $\boldsymbol{\Theta}^{(Mult)}$, may be used. This requires two sets of model parameters to be stored. It also means that the alignments used to estimate the transforms, when EM is used, may not be closely related to those for the adaptively trained system.

These problems have have limited the use of complex adaptive training schemes for unsupervised adaptation particularly for highly variable acoustic environments. In these conditions the need for a “good” estimate of the transform parameters is very important for adaptively trained systems, even more so than for multi-style trained systems. Despite these problems significant reductions in word error rate have been obtained using adaptive training on both SwitchBoard and Broadcast News Transcription tasks compared to using multi-style trained systems[1]. However to make greater use of adaptively trained systems, particularly for situations with great variability in the acoustic environment, schemes to improve the transform estimation are required.

4. ADAPTIVE TRAINING SCHEMES

It is useful to describe each of the classes of adaptive training scheme in terms of the general adaptive training theory previously given. In practice some state-of-the-art systems use a combination of adaptive training schemes. For example [11] combines the use of a model independent scheme (CMN), a feature-space transformation (VTLN) and model-space adaptation (MLLR).

4.1. Model Independent

Model independent adaptive training is the most commonly used, and simplest, form of adaptive training. It is invariably used for self-adaptation. The transform is assumed to be independent of the model parameters and the hypothesised transcription. Given these assumptions the “best” transformation is to *sphere* the data[2], i.e. transform the data so that it is zero mean and identity covariance matrix. These schemes do not require an hypothesis of the test data to be made, dramatically simplifying unsupervised adaptation. This is the basis of CMN and CVN. Since model independent schemes only alter the training data no modifications to the canonical model training algorithms are required.

There are a number of problems with model independent adaptive training schemes which limit their usefulness. The most serious limitation is that the amount of data is assumed to be large enough that the moments of the data are independent of what was said. For short blocks of data this can result in poor performance. One solution to this is to make the “moment-window” a moving average over the data. This allows the inherent variability of the transform estimate to be built into the canonical model parameters. This is the basis of RASTA processing [13]. A second limitation is that only global transforms are usually used. Many linear transformation schemes make use of multiple transforms for improved adaptation [4].

4.2. Feature-Space Transformations

Feature-space transformations may be viewed as a generalisation of model-independent adaptive training. The transformation of the features is now determined using the model parameters, $\boldsymbol{\Theta}$, and some hypothesis, $\mathcal{H}$. Feature-space transformations have the general form

$$\hat{\mathbf{o}}_t^{(s)} = \mathcal{F}^{(s)}(\mathbf{o}_t) \tag{15}$$

[1] See the NIST web-site at
http://www.nist.gov/speech/tests/ctr/h5_2001/postwshp.htm
for a series of system descriptions.

[2] Spectral subtraction [12] is not included as a model independent scheme as it requires additional information, noise estimates, to be used.

Using feature-space transformations results in very simple modifications to the standard HMM update formulae to find $\boldsymbol{\Theta}$ [3]. However due to the need to maintain consistent likelihood calculations, achieved using the Jacobian, the calculation of the transform parameters may be complicated. For biases a simple solution is possible [14]. The optimisation of a full linear matrix transformation is described in [3] (sometimes referred to as constrained MLLR) and its use in adaptive training schemes. Constrained versions of linear transforms for adaptive training [15] have also been examined. It is not necessary for there to be a single transformation for all features. Separate transforms may be associated with groups of Gaussian components from the model set, or even regions of acoustic space. A non-linear transformation scheme is VTLN [2]. The complexities of the Jacobian normalisation are usually ignored and the parameters estimated in a simple grid search fashion. In many VTLN schemes a GMM is used to estimate the warping factors, since this does not require the use of an hypothesised transcription. However for more complex transformations it is important to use the actual model parameters when estimating the transform parameters.

As the estimation of the transform parameters is dependent on the model parameters and the hypothesis it should be less sensitive to short blocks of data. However if the hypothesis is poor, such as may occur in mismatched acoustic conditions, the estimated transform may also be poor. Furthermore though more robust to short sentences than model independent schemes, there may still be large variances on the estimates of the transform parameters.

4.3. Model-Space Transformations

One of the first forms of adaptive training was model-based, gender-dependent systems. This was extended to more complex transformations in [1]. The general form of the model-based transformation is

$$\hat{\boldsymbol{\mu}}^{(sm)} = \mathcal{F}_{\mu}^{(s)}(\boldsymbol{\mu}^{(m)}), \quad \hat{\boldsymbol{\Sigma}}^{(sm)} = \mathcal{F}_{\sigma}^{(s)}(\boldsymbol{\Sigma}^{(m)}) \tag{16}$$

The component priors may also be adapted. In a similar fashion to the feature-space transformations the use of linear transforms has been extensively investigated. Full matrix transformations of the means have been used in adaptive training in [1]. Full variance transforms and the possible use in adaptive training, though with no experiments, in [3]. Recently the use of multiple cluster schemes within and adaptive training process has been proposed [16].

Model-space adaptation is the most flexible of the adaptive training schemes. The main problem with model-space adaptive training is the cost of training the models. A naive implementation requires maintaining separate statistics for every training block. For current state-of-the-art databases this is impractical. As an alternative an iterative scheme updating model means and then the variances may be used [11]. Even this iterative scheme requires significantly more memory than standard training schemes. Model-space schemes will also suffer in the same way as the feature-space transforms from limited data and poor hypothesis.

5. ADAPTIVE TRAINING EXTENSIONS

The previous sections have described the problems with using adaptive training schemes for ASR. This section describes some of the schemes that may, or have been used, in adaptive training to improve performance. As one of the main problems with adaptive training is the estimation of the target domain transform possible solutions may be taken from advances in estimating adaptation transforms for multi-style systems. The need for robust estimates is more important for adaptively trained systems than multi-style systems. Adaptively trained systems rely on the transforms to map from some normalised domain to the target domain, whereas multi-style systems are tuned from a general to a specific domain. Thus schemes that work on multi-style systems may be expected to yield greater gains for adaptively trained systems.

5.1. MAP Adaptation

Standard adaptive training uses the MLE of the transform parameters. When there is limited adaptation data, or unsupervised adaptation is being used, then the transform may be a "poor" estimate of the correct acoustic transform. One way to reduce this problem is to use priors on the transform parameters. This has been proposed for CAT [17] and MLLR [18, 19]. The estimation is now[3]

$$\hat{\boldsymbol{\lambda}}^{(\check{\mathcal{H}})} = \arg\max_{\boldsymbol{\lambda}} \left(p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \check{\mathcal{H}}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) \right) \tag{17}$$

The use of MAP adaptation may also be incorporated into the adaptive training process.

To date MAP adaptation has not been extensively used in the adaptive training framework. Given that it fits very naturally into the framework and is a simple approach to handling some of the problems this is surprising. Though MAP adaptation is useful it does not solve all the problems associated with adaptive training schemes. It does not address the need to have an initial transcription for unsupervised adaptation, nor the situation when there is very little adaptation data available.

5.2. Lattice-Based Adaptation

Recently the use of lattice-based adaptation techniques have been proposed [20] for self-adaptation. Though the experiments were based on multi-style systems the technique is directly applicable to adaptively trained schemes, particularly model-space adaptive training schemes. The following form of approximation is used

$$\begin{aligned} p(\mathbf{O}|\mathcal{M}) &= \sum_{\mathcal{H}} \int p(\mathbf{O}|\boldsymbol{\lambda}, \mathcal{M}, \mathcal{H}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) P(\mathcal{H}) d\boldsymbol{\lambda} \\ &\leq \sum_{\mathcal{H}} p(\mathbf{O}|\hat{\boldsymbol{\lambda}}^{(\mathcal{H})}, \boldsymbol{\Theta}, \mathcal{H}) P(\mathcal{H}) \end{aligned} \tag{18}$$

where

$$\hat{\boldsymbol{\lambda}}^{(\mathcal{H})} = \arg\max_{\boldsymbol{\lambda}} \left(\sum_{\mathcal{H}} p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) P(\mathcal{H}) \right) \tag{19}$$

By summing over all possible hypothesis rather than simply taking the MAP estimate, the transform parameters should be far less "tuned" to a specific hypothesis. Thus decoding with $\hat{\boldsymbol{\lambda}}^{(\mathcal{H})}$ rather than $\hat{\boldsymbol{\lambda}}^{(\check{\mathcal{H}})}$ should give a greater chance of self adaptation improving the recognition performance. Lattice-based adaptation is an iterative process since the estimation of the transform parameters is itself an EM process as HMM are being used. This form of adaptation becomes more useful as the error rate of the initial hypothesis becomes large, for example in mismatched acoustic conditions. Lattice-based adaptation allows an elegant, natural, extension to standard unsupervised adaptation. However, it does not address the issue of highly limited training data.

[3] As pointed out in [19] a true empirical Bayes estimate would use a prior distribution estimated on a separate dataset from that used to estimate the model parameters. In this paper the distribution over the transform parameters is treated simply as model parameters.

5.3. Posterior Adaptation

Both the previous schemes have relied on the use of a MAP or ML estimate of the transform parameters. To handle limited data posterior adaptation would be preferable. Here we use

$$\begin{aligned} p(\mathbf{O}|\mathcal{M}) &= \sum_{\mathcal{H}} \int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \mathcal{H}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda} P(\mathcal{H}) \\ &\leq \int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \check{\mathcal{H}}) p(\boldsymbol{\lambda}|\boldsymbol{\phi}) d\boldsymbol{\lambda} \\ &\leq \int p(\mathbf{O}|\boldsymbol{\lambda}, \boldsymbol{\Theta}, \check{\mathcal{H}}) p(\boldsymbol{\lambda}|\mathcal{M}, \mathbf{O}, \check{\mathcal{H}}) d\boldsymbol{\lambda} \end{aligned} \tag{20}$$

and $\check{\mathcal{H}}$ is given by equation 13. Rather than making a MAP, or ML, estimate of the transform parameters the posterior distribution over the transform parameters is used for the next decoding run

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}} \left(\int p(\mathbf{O}|\boldsymbol{\lambda}, \mathcal{M}, \mathcal{H}) p(\boldsymbol{\lambda}|\mathcal{M}, \mathbf{O}, \check{\mathcal{H}}) d\boldsymbol{\lambda} \right) \tag{21}$$

As previously mentioned there is no simple closed-form solution. to the integral. A simple, crude, approximation is given in [10]. This scheme used MLLR as the transformation and approximated the integral as

$$p(\mathbf{O}|\mathcal{M}, \mathcal{H}) \approx \sum_{\{\mathbf{Q}^{(\mathcal{H})}\}} \left(\prod_{t=1}^{T} \check{p}(\mathbf{o}_t|\boldsymbol{\theta}_{q_t}) \right) P(\mathbf{Q}|\boldsymbol{\theta}_d) \tag{22}$$

where, assuming a P-component prior,

$$\check{p}(\mathbf{o}|\boldsymbol{\theta}_q) = \sum_{m=1}^{M} \sum_{p=1}^{P} \check{c}^{(qmp)} \mathcal{N}(\mathbf{o}; \check{\boldsymbol{\mu}}^{(qmp)}, \check{\boldsymbol{\Sigma}}^{(qmp)}) \tag{23}$$

and $\check{\boldsymbol{\mu}}^{(qmp)} = \mathcal{E}\{\mathbf{o}|q, m, p\}$ and similarly for the covariance matrices and component priors.

The use of posterior adaptation allows the adaptively trained models to be used directly in recognition. [10] shows that this approximation is "reasonable". One important question with posterior adaptation is how to "correctly" estimate the posterior. This is not discussed in [10]. Combining posterior adaptation with lattice-based adaptation gives an elegant scheme for self, or unsupervised, adaptation.

5.4. Discriminative Adaptive Training

Recently the use of discriminatively trained models has been found to significantly reduce the error rate on large vocabulary speech recognition tasks, in particular the use of maximum mutual information estimation (MMIE) training [21]. This form of estimation has also been used in a version of adaptive training [22]. Rather than estimating the model parameters to maximise the likelihood they are trained so that

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} \left(\sum_{s=1}^{S} \log \left(\frac{p(\mathbf{O}^{(s)}|\mathcal{M}, \tilde{\mathcal{H}}) P(\tilde{\mathcal{H}}))}{\sum_{\mathcal{H}} p(\mathbf{O}^{(s)}|\mathcal{M}, \mathcal{H}) P(\mathcal{H})} \right) \right) \tag{24}$$

It would be preferable to estimate all the model parameters, the canonical model, transform prior parameters and the intermediary estimates of the transform parameters using MMIE. Though re-estimation formulae for discriminative training MLLR transforms have been proposed [23], the MMIE of canonical model parameters has not been investigated. In contrast the MMIE of canonical models with feature-space adaptive training has been investigated [22], since it requires minimal changes to the MMIE code. However in the talk presented the feature-based transforms were estimated using MLE on ML trained models and fixed for all subsequent model estimation iterations

One of the problems with full discriminative adaptive training is that supervised adaptation must be used, as the correct transcription is required to estimate the target domain transform. It also doesn't deal with the limited data problems described earlier.

6. WHERE NEXT?

This paper has given a overview of adaptive training schemes and how they may are applied in state-of-the-art systems. Various limitations and possible solutions have been described. However the possibilities for adaptive training have not been fully investigated. The use of posterior adaptation has not been examined to any extent. In particular suitable approximations are required for decoding. Furthermore for posterior adaptation to be useful appropriate prior distributions must be found. The majority of adaptive training schemes are currently based on linear transformations of the features or model parameters. However it is known that the effects of background noise for example are highly non-linear. Where there are high levels of background noise in the training data adaptive training schemes incorporating appropriate non-linear transformations may be useful e.g. PMC [24]. Techniques for MLE of the background noise conditions have already been proposed [25] but not used in the estimation of the canonical model. Finally the general framework of Bayesian networks incorporating transformations is very powerful. Little research has been performed in this area.

7. REFERENCES

[1] T Anastasakos, J McDonough, R Schwartz, and J Makhoul, "A compact model for speaker-adaptive training," in *Proceedings ICSLP*, 1996, pp. 1137–1140.

[2] L Lee and R C Rose, "Speaker normalisation using efficient frequency warping procedures," in *Proceedings ICASSP*, 1996, vol. 1, pp. 353–356.

[3] M J F Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[4] C J Leggetter and P C Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density HMMs," *Computer Speech and Language*, vol. 9, pp. 171–186, 1995.

[5] M J F Gales, "Cluster adaptive training of hidden Markov models," *IEEE Transactions Speech and Audio Processing*, vol. 8, pp. 417–428, 2000.

[6] G Zweig, *Speech Recognition with Dynamic Bayesian Networks*, Ph.D. thesis, ICSI, UC Berkely, 1999.

[7] A P Dempster, N M Laird, and D B Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[8] P C Woodland, D Pye, and M J F Gales, "Iterative unsupervised adaptation using maximum likelihood linear regression," in *Proceedings ICSLP*, 1996.

[9] M J F Gales, "The generation and use of regression class trees for MLLR adaptation," Tech. Rep. CUED/F-INFENG/TR263, Cambridge University, 1996, Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.

[10] M J F Gales, "Acoustic factorisation," in *Proceedings ASRU*, 2001.

[11] D Pye and P C Woodland, "Experiments in speaker normalisation and adaptation for large vocabulary speech recognition," in *Proceedings ICASSP*, 1997, pp. 1047–1050.

[12] S F Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions ASSP*, vol. 27, pp. 113–120, 1979.

[13] H Hermansky and N Morgan, "RASTA processing of speech," *IEEE Transactions Speech and Audio Processing*, vol. 2, pp. 578–579, 1994.

[14] A Sankar and C-H Lee, "A maximum likelihood approach to stochastic matching for robust speech recognition," *IEEE Transactions Speech and Audio Processing*, vol. 4, pp. 190–202, 1996.

[15] J McDonough, F Metze, H Soltau, and A Waibel, "Speaker compensation with sine-log all-pass transforms," in *Proceedings ICASSP*, 2001.

[16] M J F Gales, "Multiple-cluster adaptive training schemes for speech recognition," in *Proceedings ICASSP*, 2001, pp. 233–236.

[17] M J F Gales, "Cluster adaptive training for speech recognition," in *Proceedings ICSLP*, 1998, pp. 1783–1786.

[18] W Chou, "Maximum a-posterior linear regression with elliptical symmetic matrix variate priors," in *Proceedings Eurospeech*, 1999, pp. 1–4.

[19] C Chesta, O Siohan, and C-H Lee, "Maximum a-posterior linear regression for hidden markov models," in *Proceedings Eurospeech*, 1999.

[20] M Padmanabhan, G Saon, and G Zweig, "Lattice-based unsupervised MLLR for speaker adaptation," in *Proc. ISCA ITRW ASR2000*, 2000, pp. 128–131.

[21] P C Woodland and D Povey, "Very large scale MMIE training for conversational telephone speech recognition," in *Proceeding of the 2000 Speech Transcription Workshop*, June 2000.

[22] A Ljolje, "The AT&T LVCSR-2001 system," Talk given at 2001 NIST Large Vocabulary Conversational Speech Recognition.

[23] L F Uebel and Woodland P C, "Improvements in linear transforms based speaker adaptation," in *Proceedings ICASSP*, 2001.

[24] M J F Gales, "Predictive model-based compensation schemes for speech recognition," *Speech Communication*, 1998.

[25] P J Moreno, *Speech Recognition in Noisy Environments*, Ph.D. thesis, Carnegie Mellon University, 1996.