CLUSTER ADAPTIVE TRAINING FOR SPEECH RECOGNITION

M.J.F. Gales

IBM T.J. Watson Research Center, P.O. Box 218
Yorktown Heights, NY10598, USA
Email: mjfg@watson.ibm.com

ABSTRACT

When performing speaker adaptation there are two conflicting requirements. First the transform must be powerful enough to represent the speaker. Second the transform must be quickly and easily estimated for any particular speaker. Recently the most popular adaptation schemes have used many parameters to adapt the models. This limits how rapidly the models may be adapted. This paper examines an adaptation scheme requiring very few parameters to adapt the models, cluster adaptive training (CAT). CAT may be viewed as a simple extension to speaker clustering. Rather than selecting one cluster, a linear interpolation of all the cluster means is used as the mean of the particular speaker. This scheme naturally falls into an adaptive training framework. Maximum likelihood estimates of the interpolation weights are given. Furthermore, simple re-estimation formulae for cluster means, represented both explicitly and by sets of transforms of some canonical mean, are given. On a speaker-independent task CAT reduced the word error rate using very little adaptation data. In addition when combined with other adaptation schemes it gave a 5% reduction in word error rate over adapting a speaker-independent model set.

1. INTRODUCTION

In recent years there has been a great deal work on adapting speech recognition systems to both acoustic environment differences and to particular speakers. In order to adapt large numbers of parameters with very little adaptation data, some compact representation of a speaker or acoustic environment is required. There are two consideration in choosing this representation. First, the representation should be powerful enough to accurately model the speaker or acoustic environment. Second, the transform must be compact, so that its parameters may be robustly estimated using little adaptation data. A variety of representations have been examined, for example, vocal tract normalisation [4], maximum likelihood linear regression (MLLR) [5], constrained model-space transforms [3] and speaker clustering. There is also the question of the canonical model to be adapted. Originally a speaker-independent model was commonly used as the canonical model. Recently, since the majority of training databases have multiple speakers or acoustic environments, the adaptation scheme to be used in recognition has also been used during training. This is known as *adaptive training*[1]. By using adaptive training it is possible to build canonical models which only represent variability from individual speakers rather than the variability over all speakers in the training database.

This paper considers a new simple form of adaptive training, *cluster adaptive training* (CAT). The approach is related to speaker clustering. However in contrast to speaker clustering where a particular cluster is selected as the speaker model, a linear interpolation of all the clusters is used. To simplify the estimation process the form of the clusters is slightly restricted, the component weights, or priors, and variances are tied over all the speaker clusters. For any particular speaker a set of interpolation values, the *weight vector*, must be estimated. This weight vector is related to the *soft* regression class weight vectors described in [2]. Having selected the transform for a speaker, the form of the cluster means must be chosen. This paper considers two forms. The first is an explicit set of means per cluster. Alternatively, cluster dependent MLLR transforms of some canonical model may be used. In both cases simple closed-form maximum likelihood (ML) estimation formulae are given. In addition a Bayesian interpretation of the weights estimation process is described. This yields a simple posterior distribution for the weights, which may in theory be used in the recognition process, allowing instantaneous speaker adaptation.

This paper is organised as follows. The next section describes the basic form of CAT and the overall training strategy. The following section describes how the weight vectors may be estimated in an ML fashion. The estimation of both model-based and transform-based clusters will then be described. A Bayesian interpretation of the weight estimation process is also described. Results on a speaker-independent large-vocabulary task will be detailed and conclusions drawn.

2. CLUSTER ADAPTIVE TRAINING

CAT is a simple extension to the standard speaker clustering scheme. Rather than assuming that the speaker belongs to one of a set of distinct speaker classes, the speaker model parameters are determined by a linear combination of cluster means. The Gaussian component variances and weights are assumed to be the same across all speaker clusters.

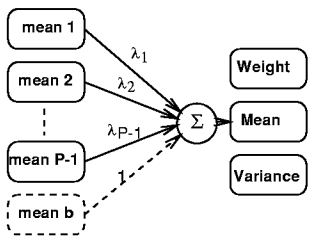

Figure 1: Cluster adaptive training

CAT is shown diagrammatically in figure 2. The op-

[1] Authors have also used the term ***adapted training***.

tional "dotted" cluster, *mean b*, always has a weight of 1. This may be used to represent any speaker-independent aspects of the mean vectors. By including a bias cluster an additional cluster may be used without increasing the number of parameters to be estimated for a particular speaker. For a particular component mean belonging to weight class r, $m \in M^{(r)}$, the speaker mean is given by

$$\hat{\mu}^{(m)} = \mathbf{M}^{(m)} \lambda^{(r)} \tag{1}$$

where there are P clusters,

$$\mathbf{M}^{(m)} = \left[\begin{array}{ccc} \mu^{(m1)} & \ldots & \mu^{(mP)} \end{array} \right] \tag{2}$$

$\mu^{(mp)}$ is the mean of Gaussian component m associated with cluster p and the extended weight vector is given by

$$\lambda^{(r)} = \left[\begin{array}{cccc} \lambda_1^{(r)} & \ldots & \lambda_{P-1}^{(r)} & 1 \end{array} \right]^T \tag{3}$$

The training scheme for CAT is similar to other adaptive training schemes [1]. The training is performed in two stages.

1. Estimate the values of the weight vectors for each speaker in the training data, given the current estimate of the cluster parameters.
2. Estimate a new set of clusters given the weight vectors.

The procedure is then repeated until some convergence criterion is satisfied. During testing a weight vector is estimated for each new speaker and a new mean calculated for decoding.

3. CLUSTER WEIGHT VECTORS

The ML estimate of the interpolation weight for each cluster is similar to transformation smoothing [2]. It can be shown that the weight vector associated with weight class r is

$$\lambda^{(r)} = \mathbf{G}_{\mathbf{w}}^{(r)-1} \mathbf{k}_{\mathbf{w}}^{(r)} \tag{4}$$

where

$$\mathbf{G}_{\mathbf{w}}^{(r)} = \sum_{m \in M^{(r)}, \tau} \gamma_m(\tau) \mathbf{M}^{(m)T} \mathbf{\Sigma}^{(m)-1} \mathbf{M}^{(m)} \tag{5}$$

$$\mathbf{k}_{\mathbf{w}}^{(r)} = \sum_{m \in M^{(r)}} \mathbf{M}^{(m)} \mathbf{\Sigma}^{(m)-1} \sum_{\tau} \gamma_m(\tau) \mathbf{o}(\tau) \tag{6}$$

and $\gamma_m(\tau)$ is the posterior probability of Gaussian component m at time τ. If the extended weight vector is used then, since there is a fixed value, only $p-1$ weights must be estimated. A similar estimation scheme is used, except that equation 6 is replaced by

$$\mathbf{k}_{\mathbf{w}}^{(r)} = \sum_{m \in M^{(r)}} \mathbf{M}^{(m)} \mathbf{\Sigma}^{(m)-1} \sum_{\tau} \gamma_m(\tau) (\mathbf{o}(\tau) - \mu^{(mP)}) \tag{7}$$

The standard "hard" speaker cluster may be simply expressed using the same expressions,

$$\lambda^{(r)} = \arg\max_{p} \left\{ k_{\mathbf{w}p}^{(r)} - \frac{1}{2} g_{\mathbf{w}pp}^{(r)} \right\} \tag{8}$$

where $\lambda^{(r)}$ is now an indicator variable to denote which cluster.

4. CLUSTER PARAMETERS

This paper considers two representation of the speaker cluster means. The first is a set of means for each cluster, the second some cluster dependent linear transformation of some canonical means. This section details the estimation scheme for both the cluster means and the cluster independent Gaussian component weights and variances. For both the model-based and cluster-based representations of the cluster, the same sufficient statistics are required to estimate the cluster parameters. These are (assuming that $m \in M^{(r)}$)

$$\mathbf{G}^{(m)} = \sum_{s,\tau} \lambda^{(sr)} \lambda^{(sr)T} \gamma_m(\tau) \tag{9}$$

$$\mathbf{K}^{(m)} = \sum_{s,\tau} \gamma_m(\tau) \lambda^{(sr)} \mathbf{o}(\tau)^T \tag{10}$$

$$\mathbf{S}^{(m)} = \sum_{s,\tau} \gamma_m(\tau) \mathbf{o}(\tau) \mathbf{o}(\tau)^T \tag{11}$$

and the number of frames assigned to that particular Gaussian component. $\lambda^{(sr)}$ is the weight vector for weight class r of speaker s. If diagonal covariance matrices are to be used, then it is only necessary to store diag $\left(\mathbf{S}^{(m)}\right)$. Diagonal covariance will be assumed for the rest of this paper. The total memory requirement for the cluster estimation for an n-dimensional feature vector, M Gaussian components and P clusters is $M(P^2 + Pn + n + 1)$ floats.

4.1. Model-based clusters

The estimation of the cluster means is a simplification of the soft MLLR transform estimation scheme described in [2]. The set of mean vectors may be estimated using

$$\mathbf{M}^{(m)T} = \mathbf{G}^{(m)-1} \mathbf{K}^{(m)} \tag{12}$$

where $\mathbf{G}^{(m)}$ and $\mathbf{K}^{(m)}$ are defined in equations 9 and 10. The variance may be shown to be

$$\mathbf{\Sigma}^{(m)} = \text{diag} \left(\frac{\mathbf{S}^{(m)} - \mathbf{M}^{(m)} \mathbf{K}^{(m)}}{\sum_{s,\tau} \gamma_m(\tau)} \right) \tag{13}$$

and the component weights are given by

$$w^{(m)} = \frac{\sum_{s,\tau} \gamma_m(\tau)}{\sum_{s,n \in S^{(m)},\tau} \gamma_m(\tau)} \tag{14}$$

where $S^{(m)}$ is the set of components in the same state as component m.

4.2. Transform-based clusters

When the clusters are to be represented as MLLR transforms of some canonical model an iterative estimation scheme is required. The mean of Gaussian component m of cluster p is now [5]

$$\mu^{(mp)} = \mathbf{A}^{(pq)} \mu^{(m)} + \mathbf{b}^{(pq)} = \mathbf{W}^{(pq)} \xi^{(m)} \tag{15}$$

where $\xi^{(m)}$ is the extended mean vector of the canonical model. Given some initial estimate of the canonical model and the fixed weight vectors associated with each speaker, the training process is as follows.

1. Estimate the set of MLLR transforms given the estimate of canonical model and the training speaker weight vectors.
2. Estimate the canonical model given the set of MLLR transforms and the weight vectors.

This is repeated until some convergence criterion is satisfied. This form of representation is related to speaker adaptive training (SAT) [1]. However in contrast to SAT a fixed number of training "speakers", or clusters, are chosen with the added flexibility of interpolating between speakers. The fixed, usually small, number of clusters has some advantages, First, the re-estimation formulae become far easier as the statistics may be stored at the cluster level. Second, the amount of data associated with each cluster is far greater than that associated with the average training speaker. It is therefore possible to use many transforms to represent the cluster. Furthermore in CAT only interpolation weights are estimated for a new speaker, rather than a new estimate of a transform as in SAT.

MLLR transforms. The estimation of the transformation parameters is similar to the linear transform optimisation described in [2]. All the transforms may be simultaneously estimated, however this involves inverting a large $P(n+1)$ square matrix per dimension. A simpler iterative approach is possible. Considering only one of the clusters, it is simple to estimate the transform parameters, given the other cluster transforms. Row i of the transform class q of cluster p is found to be

$$\mathbf{w}_i^{(pq)} = \mathbf{k}_\mathbf{t}^{(pqi)} \mathbf{G}_\mathbf{t}^{(pqi)-1} \tag{16}$$

where

$$\mathbf{G}_\mathbf{t}^{(pqi)} = \sum_{m \in M^{(q)}} g_{pp}^{(m)} \frac{1}{\sigma_{ii}^{(m)2}} \xi^{(m)} \xi^{(m)T} \tag{17}$$

$$\mathbf{k}_\mathbf{t}^{(pqi)} = \tag{18}$$
$$\sum_{m \in M^{(q)}} \frac{1}{\sigma_{ii}^{(m)2}} \left(k_{pi}^{(m)} - \sum_{j \neq p} g_{jp}^{(m)} \mathbf{w}_i^{(jq)} \xi^{(m)} \right) \xi^{(m)T}$$

Here the Gaussian components have been grouped into transform classes, which may differ from the weight classes used in the weight vector estimation. To update all the transforms, it is necessary to iterate around all the transforms, estimating each transform, given all the others. However, since $\mathbf{G}_\mathbf{t}^{(pqi)}$ is independent of the current estimate of the transform it need only be inverted once, so the process is quite efficient.

Canonical Model Estimation. There is now the issue of estimating the canonical means which are transformed by the cluster dependent transforms. The estimation of the means is slightly more complex than the model-based mean estimation and is again closely linked to the SAT re-estimation formulae. The mean is found by

$$\mu^{(m)} = \left(\sum_{i,j} g_{ij}^{(m)} \mathbf{A}^{(iq)T} \mathbf{\Sigma}^{(m)-1} \mathbf{A}^{(jq)} \right)^{-1} \tag{19}$$
$$\sum_i \mathbf{A}^{(iq)} \mathbf{\Sigma}^{(m)-1} \left(\mathbf{k}_i^{(m)T} - \sum_j g_{ij}^{(m)} \mathbf{b}^{(iq)} \right)$$

assuming that Gaussian component m is in transform class q ($m \in M^{(q)}$). Similarly the expression for the variance may be rearranged as

$$\mathbf{\Sigma}^{(m)} = \mathrm{diag} \left(\frac{\mathbf{S}^{(m)} - 2 \sum_i \mathbf{k}_i^{(m)} \mu^{(smi)T}}{\sum_{s,\tau} \gamma_m(\tau)} + \frac{\sum_{i,j} g_{ij}^{(m)} \mu^{(smi)} \mu^{(smj)T}}{\sum_{s,\tau} \gamma_m(\tau)} \right) \tag{20}$$

The Gaussian component weight estimates are the same as for the model-based cluster scheme. The estimation of the mean is thus a function of the variance and vice-versa. The optimisation of the parameters is an iterative procedure, optimising the means and then the variances and repeating.

5. BAYESIAN INTERPRETATION

To achieve very rapid adaptation, the adaptation parameters must be estimated on very little data. In these situations the ML estimate may be a poor estimate of the actual parameters. Furthermore even taking a maximum *a-posteriori* estimate of the transform parameters may be poor, since the variance on the estimates will be large. In these cases it would preferable to deal with the posterior distribution of the transform parameters rather than a single value. This involves making some assumptions about the prior distributions and the distribution of the observation parameters. Assuming that the prior distributions are Gaussian distributed and the observations are drawn from Gaussian distributed sources of variance[2] $\alpha \mathbf{\Sigma}^{(m)}$ then the posterior distribution of the transform parameters may be shown to be Gaussian with mean given by (assuming a single weight class)

$$\mu^{(\lambda)} = \left(\frac{\mathbf{G_w}}{\alpha} + \mathbf{\Sigma}^{(p)-1} \right)^{-1} \left(\frac{\mathbf{k_w}}{\alpha} + \mathbf{\Sigma}^{(p)-1} \mu^{(p)} \right) \tag{21}$$

and having variance

$$\mathbf{\Sigma}^{(\lambda)} = \left(\frac{\mathbf{G_w}}{\alpha} + \mathbf{\Sigma}^{(p)-1} \right)^{-1} \tag{22}$$

where the prior distribution has mean $\mu^{(p)}$ and variance $\mathbf{\Sigma}^{(p)}$[3]. With non-informative priors the mean of the distribution becomes the standard ML estimate as expected. Now the likelihood of the data for a given word sequence, $\mathbf{W}$, is given by

$$p(\mathbf{O}^T|\mathbf{W}) = \tag{23}$$
$$\int \sum_{\mathbf{\Theta}^T} \left(\prod_\tau p(\mathbf{o}(\tau)|\lambda, \theta(\tau)) \right) p(\mathbf{\Theta}|\lambda, \mathbf{W}) p(\lambda) d\lambda$$

where $\mathbf{\Theta}^T$ is the set of possible valid state sequences of length T and λ is the transform parameters. Although in theory this approach allows instantaneous adaptation, within the assumptions that the distributions of the transform parameters are appropriate, there are a number of issues that need to be solved. First there are no simple closed-form solutions to the calculation of the likelihood. Furthermore, to be as effective as possible sufficient statistics for the posterior distribution for *every* path hypothesised by the decoder should be stored. This dramatically increases the memory requirements during decoding. This approach will be investigated in future work.

[2] The additional term α is added to give a similar concept to the MAP τ factor.

[3] In practice the prior is liable to be multi-modal, the most obvious modes being "male" and "female".

6. RESULTS

The task used to evaluate the CAT system was an internal IBM task. It is a speaker-independent task using read speech data recorded in clean environments with the same microphone. There are 1670 speakers in the training data with a total of 36272 training sentences. A state clustered decision tree system was used throughout with 2755 states. The test set consists of 9 speakers, each uttering 61 sentences giving a total of around 10,000 words in the test set. A trigram language model was used in all tests. Two basic model sizes were considered. The first used four Gaussian components per state and the second an average of 12 Gaussian components per state. The weight vectors were estimated in a supervised adaptation mode using 20 adaptation sentences. A single weight class was used for all experiments. In preliminary results the use of many weight classes was found to give little gain in performance.

Number Gaussian Components	Number of Clusters	Word Error Rate
4	—	15.1
	1 (+1)	12.9
	8	12.3
12	—	12.6
	1 (+1)	11.7
	8	11.5

Table 1: Comparison of the baseline systems wit the CAT systems

Table 1 shows the performance of the baseline CAT systems. Two forms of model-based cluster CAT were examined. The first used two clusters with one cluster being used as a bias cluster, *1 (+1)*. Thus, there was only a single free parameter. The second used an eight cluster system with all cluster weights depending on the speaker, *8*. As expected the standard SI four Gaussian component system performed significantly worse than the 12 Gaussian component system. In both cases the use of CAT reduced the word error rate.

An alternative to the soft use of clusters the standard hard clustering scheme may be used. A four Gaussian component system with hard cluster selection was built with two clusters. The word error rate was 13.4%. This may be compared with the *1(+1)* soft clustering scheme which had a word error rate of 12.9%. As more clusters are used the differences between the hard and soft clustering should become larger.

Number Gaussian Components	Number Clusters	Type of Clusters	Word Error Rate
4	1(+1)	model	12.9
		transform	13.3
12	1(+1)	model	11.7
		transform	11.8
	8	model	11.5
		transform	11.3

Table 2: Comparison of transform and model clustering

Table 2 compares the performance of model-based and transform-based clusters. For the transform based clusters 32 full MLLR transforms, with acoustic-space clustered transform classes, were used. For small numbers of clusters the model-based schemes outperform the transform-based schemes. This is not surprising since for the 4 Gaussian components there is almost an order of magnitude reduction in the number of parameters used to represent a cluster. This reduction in the number of parameters becomes important as the number of clusters increases. Using the 12 Gaussian component system with 8 clusters, the transform-based clusters slightly out-performed the model-based clusters. This illustrates the ability to use more clusters with a transform-based clusters than model-based ones, since there are fewer free parameters to estimate from the training data.

One of advantages of CAT is that few parameters are used to adapt the models. Using only a single sentence to estimate the weight vector there was no degradation in performance for the *1(+1)* cluster system for both the 4 and 12 Gaussian component systems. However a degradation in performance was observed for the 8 cluster system. For example the 4 Gaussian component system had a word error rate of 12.9% using a single entence to estimate the weight vector. Thus, for very little adaptation data the number of clusters must be restricted, or the Bayesian interpretation used.

CAT models may also be used as the canonical models in conjunction with other transforms. The 12 Gaussian component CAT *1(+1)* system was used for adaptation with a single global constrained model-space transform and multiple MLLR transforms estimated on 50 adaptation sentences. This yielded a word error rate of 9.4%. This can be compared with using a standard SI model as the canonical model with the same linear transformation, though of course no cluster smoothing, which gave a word error rate of 10.0%. This performance gain is roughly consistent with using other adaptive training schemes.

7. CONCLUSIONS

This paper has introduced a new form of adaptive training, cluster adaptive training. The scheme uses a very simple representation of a speaker, a weight vector to smooth a set of cluster means. Two forms of cluster representation are described. The first uses sets of means themselves. The second, a far more compact representation, uses a canonical mean and MLLR transforms of that canonical mean to represent the cluster. Re-estimation formulae are given for both the weight vectors and the cluster parameters. Both forms of cluster are found to reduce the word error rate. The use of CAT in combination with other linear adaptation schemes shows gains over adapting speaker independent models.

8. REFERENCES

1. T Anastasakos, J McDonough, R Schwartz, and J Makhoul. A compact model for speaker-adaptive training. In *Proceedings ICSLP*, pages 1137–1140, 1996.
2. M J F Gales. The generation and use of regression class trees for MLLR adaptation. Technical Report CUED/F-INFENG/TR263, Cambridge University, 1996. Available via anonymous ftp from: svr-ftp.eng.cam.ac.uk.
3. M J F Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.
4. L Lee and R C Rose. Speaker normalisation using efficient frequency warping procedures. In *Proceedings ICASSP*, volume 1, pages 353–356, 1996.
5. C J Leggetter and P C Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. *Computer Speech and Language*, 9:171–186, 1995.