A MAP-LIKE WEIGHTING SCHEME FOR MLLR SPEAKER ADAPTATION

S. Goronzy, R. Kompe

Sony International (Europe) GmbH
Stuttgart Technology Center
Stuttgarter Strasse 106
70736 Fellbach, Germany
e-mail: <goronzy,kompe>@sony.de

ABSTRACT

This paper presents an approach for fast, unsupervised, on-line MLLR speaker adaptation using two MAP-like weighting schemes, a static and a dynamic one. While for the standard MLLR approach several sentences are necessary before a reliable estimation of the transformations is possible, the weighted approach shows good results even if adaptation is conducted after only a few short utterances. Experimental results show that using the static approach can improve the word error rate by approx. 27% if adaptation is conducted after every 4 utterances (single words or short phrases). Using the dynamic approach, results can be improved by 28%. The most important advantage of the dynamic weight is that it is rather insensitive with respect to the initial weight whereas for the static approach it is very critical which initial weight to chose. Moreover, useful values for the weights in the static case depend very much on the corpus. If the standard MLLR approach is used, even a drastic increase in sentence error rate can be observed for these small amounts of adaptation data.

1. INTRODUCTION

State of the art speaker-dependent (SD) systems yield much higher recognition rates than speaker independent (SI) ones. However for many applications, in particular consumer devices, it is not feasible to gather enough data from one speaker to train the system. To overcome this mismatch in recognition rates, speaker adaptation algorithms are widely used in order to achieve recognition rates that come close to SD systems but only use a fraction of speaker dependent data compared to SD systems.

In the past, Maximum Likelihood Linear Regression (MLLR) has proven to be quite effective for speaker adaptation of Continuous Density Hidden Markov Models (CDHMMs) [1, 2, 3]. The basic principle is to adapt the parameters of the Gaussian densities, the basis for which is a set of previously trained SI models, by calculating one or several transformation matrices from the speaker specific data (the speech data received from the new speaker while (s)he is using the system, henceforth adaptation data). Its major advantage in comparison to other adaptation methods is its ability to update even the parameters of those Gaussian distributions that have not been observed in the adaptation data.This is achieved by clustering several Gaussians, so that they build regression classes that share the same transformation or regression matrix. A high degree of clustering can thus reduce the number of parameters to be estimated. In case there is only a very limited amount of adaptation data available, only one global transformation matrix is calculated. This is very fast, but adaptation is rather coarse. If more adaptation data is available, two or more regression classes can be used which then become more and more specific. However, with an increasing number of regression classes more adaptation data is necessary to be able to reliably calculate the transform for each regression class. It has been reported in [2] that several sentences spoken by the new speaker are necessary before the HMM parameters can be adapted successfully. If less data is available, the transformation matrices may be ill-conditioned due to the lack of data and this may cause a decrease in recognition rate.

For Command&Control systems a faster approach is needed, using unsupervised, on-line mode, that means the spoken words are not known and the models are adapted continuously while the system is in use. For speech controlled consumer devices it is not desirable for the user to undergo a training procedure before (s)he can use the system. This is in contrast to dictation systems, where the user has to read a long text that is used for adaptation before (s)he can actually use the system.

For this purpose we introduce a MAP-like weighting scheme cf. [5] for MLLR adaptation that can either be static or dynamic and that is capable of rapid adaptation on very small amounts of data, such as single words. To achieve this, the adapted mean[1] is not calculated as in the standard MLLR approach, but instead a weighted sum of the previous mean and the new mean transformed by MLLR is taken.

Using the static approach the problem arises that oscillation in HMM parameters is very high, even if the system is used for a long time by the same speaker. The recognition rates do not really converge to an optimum. As a further consequence, a series of misrecognitions at such a late stage may cause a deterioration in recognition rates. To avoid this, the dynamic scheme is applied. Utterances that were spoken a long time ago by the same speaker then still have a strong influence.

2. THE MLLR APPROACH

In the standard MLLR approach [1, 2], the mean vectors μ of the Gaussian densities are updated using a $n \times (n+1)$ transformation matrix $\mathbf{W}$ calculated from the adaptation data by applying

$$\bar{\mu} = \mathbf{W}\hat{\mu} \tag{1}$$

where $\hat{\mu}$ is the extended mean vector:

$$\hat{\mu} = \begin{bmatrix} \omega \\ \mu_1 \\ . \\ . \\ \mu_n \end{bmatrix} \tag{2}$$

[1] Also other parameters, such as variances and mixture weights, can be adapted, but in this paper only the means are considered.

ω is the offset term for regression ($\omega = 1$ for standard offset, $\omega = 0$ for no offset), n is the dimension of the feature vector. For the calculation of the transformation matrix $\mathbf{W}$, the objective function to be maximized is the likelihood of generating the observed speech frames:

$$F(O|\lambda) = \sum_{\theta \in \Theta} F(O,\theta|\lambda). \quad (3)$$

By maximizing the auxiliary function

$$Q(\lambda,\bar{\lambda}) = \sum_{\theta \in \Theta} F(O,\theta|\lambda) log(F(O,\theta|\bar{\lambda})), \quad (4)$$

where

O	is a stochastic process, with the adaptation data being a series of T observations generated by this process,
λ	is the current set of model parameters,
$\bar{\lambda}$	is the re-estimated set of model parameters, and
θ	is the sequence of states to generate O,

the objective function is also maximized. Iteratively calculating a new auxiliary function with refined model parameters will therefore iteratively maximize the objective function [4] unless it is a maximum. Maximizing equation (4) with respect to $\mathbf{W}$ yields in the case of Gaussian densities

$$\sum_{t=1}^{T} \gamma_s(t) \mathbf{C}_s^{-1} o_t \hat{\mu}_s' = \sum_{t=1}^{T} \gamma_s(t) \mathbf{C}_s^{-1} \mathbf{W} \hat{\mu}_s \hat{\mu}_s', \quad (5)$$

where

s	is the current state,
$\gamma_s(t)$	is the total occupation probability of state s at time t,
$\mathbf{C}_s^{-1}$	is the inverse covariance matrix of the Gaussian density that is assigned to s at time t and
o_t	is the observed feature vector at time t.

2.1. Tied Regression Matrices

Since we have one (or several) regression matrice(s) shared by several Gaussians, the summation has to be performed over all of these Gaussians, i.e. for the non-mixture case,

$$\sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{s_r}(t) \mathbf{C}_{s_r}^{-1} o_t \hat{\mu}_{s_r}' = \sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{s_r}(t) \mathbf{C}_{s_r}^{-1} \mathbf{W}_l \hat{\mu}_{s_r} \hat{\mu}_{s_r}', \quad (6)$$

where R is the number of Gaussians sharing one regression matrix and l the regression class index. Eq. (6) can be rewritten as

$$\sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{s_r}(t) \mathbf{C}_{s_r}^{-1} o_t \hat{\mu}_{s_r}' = \sum_{r=1}^{R} \mathbf{V}_r \mathbf{W}_l \mathbf{D}_r, \quad (7)$$

where $\mathbf{V}_r$ is the state distribution inverse covariance matrix scaled by the state occupation probability and $\mathbf{D}_r$ is the outer product of the extended means.

The left hand side of equation (6) is further denoted as $\mathbf{Z}_l$, which is an $n \times (n+1)$ matrix. If the individual matrix elements of $\mathbf{V}_r$, $\mathbf{D}_r$ and $\mathbf{W}_l$ are denoted by $v_{r,ij}$, $d_{r,ij}$ and $w_{l,ij}$ respectively we get after further substitutions

$$z_{l,ij} = \sum_{q=1}^{n+1} w_{l,iq} g_{i,jq}, \quad (8)$$

where $g_{i,jq}$ are the elements of an $(n+1)(n+1)$ matrix $\mathbf{G}_i$. $z_{l,ij}$ and $g_{i,jq}$ are not dependent on $\mathbf{W}_l$ and can be computed from the observation vector and the model parameters:

$$w_{l,i}' = \mathbf{G}_i^{-1} z_{l,i}' \quad (9)$$

$w_{l,i}$ and $z_{l,i}$ are the i^{th} rows of $\mathbf{W}_l$ and $\mathbf{Z}_l$ respectively and so $\mathbf{W}_l$ can be calculated on a row by row basis.

2.2. Implementation Issues

Since a fast approach is needed in the case of consumer devices, we use the Viterbi algorithm for adaptation, where each speech frame is assigned to exactly one state, so that

$$\gamma_{s_r}(t) = \begin{cases} 1 & \text{if } \theta_t = s_r \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Therefore along the best path, Eq. 6 becomes in the case of one regression class

$$\sum_{t=1}^{T} \mathbf{C}_{s_r}^{-1} o_t \hat{\mu}_{s_r}' = \sum_{t=1}^{T} \mathbf{C}_{s_r}^{-1} \mathbf{W} \hat{\mu}_{s_r} \hat{\mu}_{s_r}', \text{with } s_r = \theta_t \quad (11)$$

It should be noted that in all our experiments we used diagonal covariance matrices. In case of mixture Gaussians, we only use the density with the highest probability for each frame, so that instead of the optimal state sequence we are using the optimal sequence of Gaussians and the same formula (Eq.11) can be applied, only the s_r denote the states in one case and the Gaussians in the other one. In all experiments one maximization iteration was used. Since only results for 1 regression class are reported in this paper, the assignment of Gaussians to different regression classes will not be discussed here.

2.3. Weighting of the Adapted Means

2.3.1. Using a Static Weighting Factor

This approach differs from the one described above, in that the adapted mean is not calculated as described there, but an additional weighting factor α is introduced. A weighted sum of the 'old'(as calculated in the previous adaptation step) and 'new' mean (as calculated in Eq. 1 for the current adaptation step) is used. The update of the means is therefore performed according to Eq. 12 for the static approach:

$$\hat{\mu}_j^k = \alpha \vec{\mu}_j^k + (1-\alpha) \hat{\mu}_j^{k-1} \quad (12)$$

where k denotes the current adaptation step. $\vec{\mu}_j^k$ is the mean estimated from the utterance of the current adaptation step k. The value that was chosen for α turned out to have a very strong influence on the adaptation results. Also the optimal value of α very much depended on the number of speech frames that were used for calculating the transformation matrix, the number of regression classes and the corpus used for testing. In an application where we eventually want to switch between different vocabularies and may want to modify these at a later stage, it is not feasible to always look for and find the optimal weighting factor. A solution to this problem is the use of a dynamically changing weighting factor as described in 2.3.2, where the adaptation performance is relatively independent of the initial weight chosen. Another problem is that with weighting each utterance equally the change in means is relatively high each time the adaptation is applied. As a consequence the speaker adapted models may be 'destroyed' at a late stage of adaptation if e.g. a series of misrecognitions occurs. Again this problem can be solved by using the dynamic weighting scheme.

2.3.2. Using a Dynamically Changing Weighting Factor

To avoid the high oscillation in HMM parameters mentioned in the previous section, a dynamic scheme, that was inspired by on-line MAP adaptation, cf. [5], was applied. Major changes are made to the SI models when a new speaker starts using the system, but they become smaller the longer the speaker is using it. That means the weight for the actual utterance slowly decreases over time. This is achieved by taking into account

the number of frames that have already been observed from that specific speaker for that specific regression class in the current and all previous adaptation steps. For the dynamic scheme the weights α_l for each regression class l are then calculated as follows:

$$\alpha_l^k = \frac{n_l^k}{\tau_l^{k-1} + n_l^k} \tag{13}$$

$$(1-\alpha_l^k) = \frac{\tau_l^{k-1}}{\tau_l^{k-1} + n_l^k} \tag{14}$$

where n_l^k in general is

$$n_l^k = \sum_{t=1}^{T} \sum_{r=1}^{R} \gamma_{s_r}(t) \tag{15}$$

In our case (Eq.10) it is a counter of the number of frames that were so far observed in the regression class l. τ_l is a weighting factor for that class, the initial value of which will be determined heuristically. Now Eq. 12 can be rewritten as

$$\hat{\mu}_j^k = \frac{\tau_l^{k-1}\hat{\mu}_j^{k-1} + n_l^k \bar{\mu}_j^k}{\tau_l^{k-1} + n_l^k} \tag{16}$$

where τ_l increases by n_l^k after each adaptation step:

$$\tau_l^k = \tau_l^{(k-1)} + n_l^k \tag{17}$$

Using this weighting scheme, α_l and thus the influence of the most recent observed means decreases over time and the parameters will move towards an optimum for that speaker. A further advantage of this dynamic scheme is, that if already a set of speaker adapted models was obtained, but nevertheless misrecognitions occur, they do not have such a big effect on the adapted models anymore and thus a 'destruction' of the adapted models by a series of misrecognitions can be avoided. Of course, as in any unsupervised adaptation, 'many' misrecognitions in the beginning may also cause a deterioration in recognition rates.

3. EXPERIMENTAL SETUP

Training and testing were conducted using a German data base, recorded in our noise free studio and it mainly consists of isolated words and short phrases. The speech was sampled at 16 kHz and coded into 25.6ms frames with a frame shift of 10ms. Each speech frame was represented by a 39-component vector consisting of 12 MFCC coefficients and energy and their corresponding first and second time derivatives. In all adaptation experiments only the MFCC components and energy were adapted to reduce the computational cost. Furthermore, it was found that adapting the time derivatives with small amounts of data may have an adverse effect [2].

A set of speaker-independent monophone models was trained using 50 speakers (21434 utterances). We used 3-state HMMs with 1 Gaussian per state. This was due to the memory and speed requirements in the Command&Control domain. The test set consisted of 15 speakers. All results shown in this paper are averaged over all 15 speakers. Two corpora, 'a' and 'k', were used for testing. They consist of German addresses (corpus 'a', street and city names) and German commands for all kinds of consumer devices (corpus 'k', like e.g., start, rewind, switch on the TV, I would like to see the movie with Clint Eastwood, etc.). 'a' comprises on the average 23 utterances per speaker (approximately 1.2 words per utterance), 'k' comprises on the average 234 utterances per speaker (approximately 2 words per utterance).

	a	k
SI	6.4	11.5
MLLR	10.1	15.8
MLLR+static weight	4.6	8.4
MLLR+dynamic weight	4.6	8.3

Table 1: % SERs using the SI system and MLLR adaptation with 1 regression class, a minimum number of speech frames of 1000 and no, static and dynamic weighting factor

These SI models were used as the starting point for the speaker adaptation in all experiments. The adaptation was done incrementally, i.e., after having recognized n speech frames[2], the optimal state sequence was computed for each utterance, and the transformation matrices were calculated. Then the means were updated. The model set using these updated means was then used for recognizing the following utterance and so on. The recognition results were obtained during this adaptation procedure. For each speaker this process was restarted using the SI models.

4. RESULTS

Table 1 shows the results for the different methods discussed above in % sentence error rate (SER). The relatively poor recognition rates of the SI system result from the fact that only monophone models are used. All experiments used 1 regression class and adaptation was conducted after every 1000 speech frames (10s of speech, what in our case corresponds to approximately 4 utterances). So the transformation matrix was each time calculated using the 1000 previously observed speech frames only. It can be seen that for the standard MLLR approach a deterioration in recognition rates can be observed (an increase in SER of 59% and 38% for 'a' and 'k' respectively). This effect was also mentioned in [1] for small amounts of adaptation data. Using the static weight, an improvement in SER of around 27% for both 'a' and 'k' can be observed. Results for the dynamic weight achieve the same improvement for 'a', the results for 'k' are slightly better (28% compared to the SI system). In general it can be seen that for such small amounts of adaptation data, the standard MLLR approach cannot be used, but both weighted MLLR approaches drastically improve the SER. It is remarkable that even for 'a' where only few utterances per speaker are available a SER reduction of 27% could be achieved, although for the first utterances there is no or little adaptation conducted.

The static weighting factors yielding the best performance turned out to depend very much on the data base, the number of regression classes, the corpus and the minimum number of frames used for adaptation. Several experiments were necessary to find the optimal factor. In the results shown below, α was set to 0.6 for 'a' and to 0.2 for 'k'. The initial value of τ_l for the dynamic case was set to 1000 for 'a' and 'k' respectively. Several experiments showed that the initial value of τ_l only has very little effect on the adaptation performance. Therefore, even though the improvements are comparable for both the static and the dynamic weight, the big advantage of the dynamic weighting scheme is the fact that the initial value of the weight is not very important. The performance was much more sensitive to the value of the static weight and in some cases more than one local maximum could be observed, so that for each corpus and each minimum number of frames and each database many experiments had to be conducted to find the optimal value for the static α. Of course this is not usable for a system that is going

[2] If the number of n speech frames was reached, adaptation was not carried out immediately, but at the end of the utterance.

freq. of adaptation	a	range	k	range
1 utterance	4.9	4.9-37.7	11.7	11.7-67.3
2 utterances	4.6	4.6-7.8	9.0	9.0-9.8
4 utterances	4.6	4.6-6.7	8.3	8.3-8.8

Table 2: % SERs and range of SER using the dynamic weight for MLLR adaptation after about 1, 2 and 4 utterances

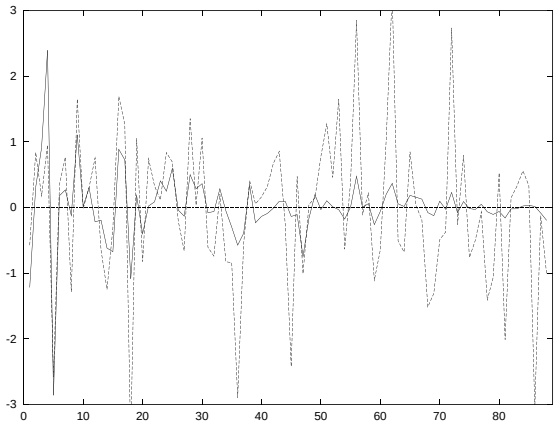

Figure 1: Change of one arbitrarily chosen mean component (y-axis) after n adaptation steps (x-axis) for static (dashed line) and dynamic (solid line) weighting.

to be used to control different applications. For such kind of applications, adaptation has to be rather insensitive to the initial weighting factors.

Further experiments were conducted using the dynamic scheme but less speech frames. Table 2 shows the results in SER for adaptation after about 1, 2 and again 4 utterances and it also shows the range of the recognition rates when using different initial values of τ_l. One can see, that for 1 utterance, the result for 'k' is a bit worse than for the SI system; choosing the wrong initial value for τ_l can cause a deterioration in recognition rates for both 'a' and 'k'. But when using 500 speech frames (about 2 utterances), drastic improvements for both corpora can already be observed and these can be further improved in the case of 'k' by using 4 utterances. Looking at the results in more detail, one can see that the range in SER for different initial values of τ_l is rather big for 1 utterance. For '4 utterances' the results seem to be quite stable. The range of the initial τ_l was varied between 100 and 10,000, without a significant change in SER. In contrast the static weight was varied between 0.09-0.9 and a drastic increase in SER could be observed when changing it by 75% from the optimum. This is an important point especially if a dynamic vocabulary is going to be used. Then it is of course not feasible to use a weighting factor for adaptation that depends that much on the vocabulary as the static one does.

In Figure 1 the change of one arbitrarily chosen component of a mean vector is shown after each adaptation step, for the static and the dynamic weight, respectively. The change is defined as the difference between the current and the previous value of that mean component. While the change for the static weight is rather big even after many adaptation steps, it slowly decreases if the dynamic weight is used. In this example adaptation was conducted on corpus 'k' after every 1000 speech frames and the means were updated 88 times in total.

5. CONCLUSION

A method for rapid, unsupervised and on-line MLLR adaptation using a static and a dynamic, MAP-like weighting scheme, respectively, has been presented in this paper. By the use of both weighting schemes, fast adaptation to new speakers is conducted with very little adaptation data, such as single words. Improvements of 27% and 28% in SER can be achieved for the static and the dynamic scheme respectively using 1 regression class and conducting adaptation each time 1000 frames have been processed. The standard MLLR approach causes an increase in SER for the considered amounts of adaptation data. We expected that the dynamic approach would outperform the static one, but the SERs are almost the same. Still the dynamic approach is superior to the static one since it turned out to be much less sensitive to the choice of the initial weight if a minimum number of 1000 speech frames is used for calculating the transformation. So it is well suited for systems where different Command&Control applications are handled by the same speech recognizer. For these applications the static weight is not applicable because it strongly depends on various factors, such as vocabulary, number of speech frames used etc. Additionally in the dynamic approach major changes are made to SI models in the beginning but only fine tuning is done if the system is used by the same speaker for a longer time. Thus misrecognitions at a late stage do not cause deterioration in performance as it could be the case when using the static weighting scheme where each utterance has the same influence on the adaptation.

6. FUTURE WORK

The results presented in this paper only used one regression class for MLLR adaptation, more experiments should be conducted using a higher number of regression classes. Nevertheless, more computational and memory effort is then needed. For the static weight a lot of additional experiments, e.g. using mixture Gaussians or triphones have been conducted and also show encouraging results. However, due to computational and memory requirements in Command&Control applications a 'simple' system using only monophones and 1 regression class could be better suited. Since the adaptation is then rather coarse a combination with MAP adaptation, which specifically adapts single phonemes but therefore needs rather big amounts of adaptation data could be beneficial. Preliminary results using a combination of these two adaptation schemes, where MAP adaptation is conducted as soon as enough adaptation data becomes available are very promising.

7. REFERENCES

[1] C.L. Leggetter,P.C. Woodland: Speaker Adaptation of HMMs using Lin. Regression, *Technical Report*, Cambridge Univ., 1994.

[2] C.L. Leggetter: Improved Acoustic Modeling for HMMs using Linear Transform., *PhD Thesis*, Cambridge Univ., 1995.

[3] C.L. Leggetter, P.C. Woodland: Maximum Likelihood Lin. Regression for Speaker Adaptation of Continuous Density HMMs, *Computer Speech and Language, pp. 171-185*, 1995.

[4] L.E. Baum: An Inequality and Associated Maximization Technique in Statistical Estimation for Prob. Functions of Markov Proc., *Inequalities*, Vol. 3, pp. 1-8, 1972.

[5] Q. Huo and C-H. Lee: On-line Adaptive Learning of the CDHMM based on Approximate Recursive Bayes Estimate, IEEE Trans. on SAP, 5(2), pp. 161-172, March 1997.