# A NOVEL TARGET-DRIVEN MLLR ADAPATATION ALGORITHM WITH MULTI-LAYER STRUCTURE

*Jia Lei* *Xu Bo*

**Institute of Automation,**
**Chinese Academy of Science,**
**Beijing 100080**
{ljia, xubo}@nlpr.ia.ac.cn

## ABSTRACT

This paper presents a novel target-driven MLLR adaptation algorithm with multiply layer structure, which is based on the thorough analysis of MLLR using the generation of regression class trees. The new algorithm is constructed on the target-driven principal. It generates the regression class dynamically, basing on the outcome of the former MLLR transformation. The regression classes is defined in order to have the maximizing increase of the auxiliary function, which is in proportional to the likelihood of the occurrence of the adaptation data. Because of the new algorithm's special transformation structure, computation load in performing transformation is much reduced. In comparison with the conventional MLLR using the generation of regression class trees, the new algorithm give a further error reduction 10% and has only half computation time consuming.

## 1. INTRUDUCTION

Speaker adaptation techniques try to adapt the initial speaker independent system (SI) to obtain near speaker dependent (SD) performance with only small amounts speaker specific data. Many adaptation techniques is developed for this aim, among which MAP[1], MLLR[2] and MLLR using the generation of regression trees[3,4] have made some progress. MAP estimation uses a prior distribution and can get robust parameter estimates in less data compared to MLE estimation. Although MAP estimation can convergence to the MLE estimation as the adaptation data increase, its adaptation is slow for it only updates distributions for which observations occur in the adaptation data. MLLR estimation is applied in order to capture the general relationship between the speaker independent modal set and the current speaker. A global linear transform matrix is estimated in [2] in order to maximize the likelihood of the occurrence of the adaptation data, then all mean parameters of the system are transformed by this matrix. By using a regression class trees[3,4], MLLR can be applied to a set of transform classes in which some output distributions of the HMM parameter set is transform tied together, based on the assumption that all the output distributions close together in the acoustic space should be tied and transformed together. A large improvement is obtained over conventional MLLR[3] by the using of regression classes.

MLLR using a regression class trees have the drawback of making the above assumption. This assumption is not right in some cases that the test speaker's acoustic property is much different from the speakers in the acoustic modal training set. In this paper, a novel algorithm is presented in order to find more suitable regression classes to improve recognition accuracy. This algorithm defines the regress classes on the augment of the auxiliary function, which is in proportional to the likelihood of the occurrence of the adaptation data. High accuracy can be obtained in this way. In addition to this, the new algorithm based its current MLLR on the former MLLR transformation. The special multi-layer transformation structure makes this algorithm have much less computation complexity.

## 2. MLLR USING THE GENERATION OF REGRESSION CLASS TREES

### 2.1 MLLR

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

$$\hat{\boldsymbol{\mu}}_{ijk} = \mathbf{W}\boldsymbol{\xi}_{ijk} \tag{1}$$

Here $\boldsymbol{\xi}_{ijk} = (1, \boldsymbol{\mu}_{ijk}) = (1, \mu_{ijk}(1), \mu_{ijk}(2), \cdots \mu_{ijk}(n))^{\mathrm{T}}$ is the extended mean vector. $\boldsymbol{\mu}_{ijk}$ is kth mean vector for each HMM's transition from state $i$ to $j$. $\mathbf{W}$ is the set of transform matrix. $n$ is the dimension of the feature vector. For the calculation of the transform matrix $\mathbf{W}$, the objective function to be maximized is the likelihood of generating the observed speech frames:

$$Q_b = \sum_i \sum_j \sum_{k=1}^{M} Q_{b_{ij}}(\lambda, \mathbf{b}_{ijk}) \tag{2}$$

$$Q_{b_{ij}}(\lambda, \mathbf{b}_{ijk}) = -\frac{1}{2} F(\mathbf{O}\,|\,\boldsymbol{\lambda}) \sum_t^T \gamma_t(i,j,k)[K_{ijk(t)} + \log(|\boldsymbol{\Sigma}_{ijk(t)}|) + (\mathbf{o}(t) - \hat{\boldsymbol{\mu}}_{ijk(t)})^{\mathrm{T}} \boldsymbol{\Sigma}_{ijk(t)}^{-1} (\mathbf{o}(t) - \hat{\boldsymbol{\mu}}_{ijk(t)})] \tag{3}$$

where

$\mathbf{O}$ is a stochastic process, with the adaptation data being a series of T observations generated by this process. $\boldsymbol{\lambda}$ is the current set of modal parameters. $F(\mathbf{O}\,|\,\boldsymbol{\lambda})$ is the likelihood of generating the observed speech frames. $\gamma_t(i,j,k)$ is the probability of taking transition from state $i$ to state $j$, being the $k^{th}$ component of the output.

Define $\mathbf{c}$ as the set of output distributions tied and transformed together. $\mathbf{W}_c$ is the transform matrix of this set. The solution of the above problem is shown in (4)-(8):

$$\mathbf{Z} = \sum_{t=1}^{T} \sum_{b_{ijk} \in \mathbf{C}} \gamma_t(i,j,k) \boldsymbol{\Sigma}_{ijk}^{-1} \mathbf{o}_t \mathbf{m}_{ijk(t)}^{\mathrm{T}} \tag{4}$$

$$\mathbf{V}_{ijk} = \sum_{t=1}^{T} \gamma_t(i,j,k) \boldsymbol{\Sigma}_{ijk(t)}^{-1} \tag{5}$$

$$\mathbf{D}_{ijk} = \mathbf{m}_{ijk} \mathbf{m}_{ijk}^{\mathrm{T}} \tag{6}$$

$$\mathbf{G}_q = \sum_{b_{ijk} \in \mathbf{C}} v_{qq} \mathbf{D}_{ijk} \tag{7}$$

$$\mathbf{W}_{c,q} = \mathbf{Z}_q \mathbf{G}_q^{-1} \tag{8}$$

where $v_{qq}$ is the $q^{th}$ element of the vector $\mathbf{V}_{ijk}$. $\mathbf{W}_{c,q}$ denotes the $q^{th}$ rows of transform matrix $\mathbf{W}_c$.

### 2.2 MLLR using the generation of regression class trees

In order to maximize the use of the small amount of adaptation data, a regression class trees is used in [3]. At the beginning of adaptation, V*iterbi*-alignment is used in order to allocate each frame of the adaptation data to each output distribution. To accumulate the statistics for the adaptation process, the summed state occupation probability and the observation vectors associated with each output distribution are recorded. A search is then made through the tree starting at the root nodes to the leaf nodes to define sets of the regression classes, which has sufficient adaptation data to get a robust estimation of the transform matrix. Each regression class in the set is then transformed individually.

## 3. TARGET-DRIVEN MLLR WITH MULTIPLY LAYER STRUCTURE

### 3.1 Define regression classes

Let's define a set of the regression classes $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots \mathbf{c}_l\}$ first. Each element $\mathbf{c}_m (1 \le m < l)$ represents a regression class, with which a single transformation matrix $\mathbf{W}_m (1 \le m \le l)$ is associated. How to define the set $\mathbf{C}$ in order to have the maximizing increment of the target function is the key problem in this paper.

According to our problem, the target function $Q_b$ in 2.1 can be change to the problem of finding the best regression classes $\mathbf{c}_m (1 \le m < l)$ and the transform matrix $\mathbf{W}_m (1 \le m \le l)$ in order to have the maximum value of the target function $Q_b$:

$$Q_b = \sum_{b_{ijk} \in C_1} \sum_{j} \sum_{k=1}^{M} Q_{b_{ij}}(\lambda, \mathbf{b}_{ijk}) + \sum_{b_{ijk} \in C_2} \sum_{j} \sum_{k=1}^{M} Q_{b_{ij}}(\lambda, \mathbf{b}_{ijk}) + \cdots + \sum_{b_{ijk} \in C_l} \sum_{j} \sum_{k=1}^{M} Q_{b_{ij}}(\lambda, \mathbf{b}_{ijk}) \tag{9}$$

Put the (3) into (9) and omit the items which have no relationship with the transform matrix $\mathbf{W}_m (1 \le m \le l)$ in (9), and we can get the new target function $\tilde{Q}_b$. Find the optimal $\mathbf{c}_m (1 \le m < l)$ and $\mathbf{W}_m (1 \le m \le l)$ to minimum the $\tilde{Q}_b$ will be our problem now.

$$\tilde{Q}_b = \min_{\substack{\mathbf{c}_m \in \mathbf{C}, 1 \le m \le l \\ W_m \in \boldsymbol{\omega}, 1 \le m \le l}} \Big( K_{c_1} \sum_{i} \sum_{j} \sum_{k=1}^{M} Trans(b_{ijk}) + K_{c_2} \sum_{i} \sum_{j} \sum_{k=1}^{M} Trans(b_{ijk}) + \cdots + K_{cl} \sum_{i} \sum_{j} \sum_{k=1}^{M} Trans(b_{ijk}) \tag{10}$$

where

$$Trans(b_{ijk}) = \sum_{t=1}^{T} [\gamma_t(i,j,k) \times (\mathbf{o}(t) - \mathbf{W}\boldsymbol{\xi}_{ijk(t)})^{\mathrm{T}} \boldsymbol{\Sigma}_{ijk(t)}^{-1} (\mathbf{o}(t) - \mathbf{W}\boldsymbol{\xi}_{ijk(t)})] \tag{11}$$

The above problem can be interpreted as the problem of transforming the current value of mean parameter in the regression class to the observed vector by a linear transformation. To define the linear regression class by the assumption, which is made in the MLLR using the generation of regression trees[3], is not always suitable because the closeness of the distribution parameters in the acoustic space does not guarantee that the observed vector should be together. So the tying of these parameters can't always ensure a good transformation. The assumption does more harms to the transformation especially when the testing speaker's acoustic feature is much different from that of the speakers in the independent acoustic modal training set.

The new algorithm is constructed mainly to overcome the drawback of the above assumption.

Let's define some base regression classes, which can be got by using a simple clustering of the original output distributions. The final regression classes $\mathbf{c}_m (1 \le m < l)$ are all made up of these base regression classes. We denote the base regression classes set as $\mathbf{C}_{base} = \{\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \cdots, \tilde{\mathbf{c}}_{num}\}$. Define $Tar_0(\tilde{c}_i)$ and $Tar_1(\tilde{c}_i)$ as the target function of the base regress class $\tilde{c}_i$ before and after adaptation respectively.

$$Tar(\tilde{c}_i \mid \tilde{c}_i \in \mathbf{C}_{base}) = \sum_{i} \sum_{j} \sum_{k=1}^{M} Trans(b_{ijk}) \tag{12}$$

$$Tar_0(\tilde{c}_i) = Tar(\tilde{c}_i \mid \tilde{c}_i \in \mathbf{C}_{base}, \mathbf{W}_m = [0:\mathbf{I}]) \tag{13}$$

$$Tar_1(\tilde{c}_i) = Tar(\tilde{c}_i \mid \tilde{c}_i \in \mathbf{C}_{base}, \mathbf{W}_m) \tag{14}$$

The algorithm finds the regression classes from the base regression class set one by one in an iteration manner. The procedure is listed below:

1.First a global MLLR transformation $\mathbf{W}_{all}$ is performed on the base regression classes set $\mathbf{C}_{base}$. Calculate the target function increase rate of each $\tilde{c}_i$ in $\mathbf{C}_{base}$:

$$Rate(\tilde{c}_i \mid \tilde{c}_i \in \mathbf{C}_{base}) = (Tar_1(\tilde{c}_i) - Tar_0(\tilde{c}_i)) / Tar_0(\tilde{c}_i) \tag{15}$$

2.Sort the target function increase rate sequences $Rate(\tilde{c}_i \mid \tilde{c}_i \in \mathbf{C}_{base})$ and define the base regression classes with the top *fixNum* biggest target function increase rate as the chosen regression class set $\mathbf{C}_c$, where *fixNum* is a fix number that can be adjusted according to the amounts of adaptation data. The more adaptation data we get, the smaller *fixNum* is.

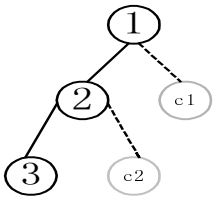
3.After the definition of a regression class set $\mathbf{C}_c$, a global MLLR associated with this set can be performed. Finally renew the base regression classes set $\mathbf{C}_{base}$ by eliminating the base regression classes which have been transformed in set $\mathbf{C}_c$.

4.Continue another regression class set $\mathbf{C}_c$ definition process until the base regression classes number in $\mathbf{C}_{base}$ is little than a fix number.

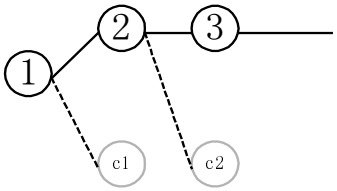
### 3.2 The efficient calculation structure

The new algorithm defines the current regression class according to the outcome of the former MLLR transformation, then transformed the regression class. After that, renew the base regression classes sets and begin another regression class's definition process. It is the special transformation structure that makes the computation complexity much less.

Figure 2 Transform structure of the MLLR using the generation of regression class trees.




Fingure3 Transform structure of the target-driven MLLR with multiply layer structure




Here fingure2 illustrate the transformation process of MLLR using the generation of regression classes trees. When a regression class, for example $\mathbf{c}1$ which is represented in gray circle in fingure2, is defined, its MLLR transformation must be calculated form the its father node 1. It is the same case with regression class $\mathbf{c}2$. Heavy computation burden can be involved because the father node may contain much more output distributions than that the regression class to be transformed contains.

Fingure3 illustrates the transformation process of the new algorithm. After a regression class $\mathbf{c}1$ is defined, the base regression classes in that class can be eliminated from the current base regression classes set. Then the following regression class definition and MLLR transformation can be performed in the eliminated base regression classes set. In the processes of defining the current regression class, we needn't calculation the $\mathbf{G}_q(7)$ of the current $\mathbf{C}_{base}$ using the output distributions in this set. We can simply subtract the $\mathbf{G}_q$ of the former MLLR regression class $\mathbf{C}_c$ form the former $\mathbf{C}_{base}$ to get the $\mathbf{G}_q$ of the current $\mathbf{C}_{base}$. Much faster calculation speed is obtained by this subtraction.

## 4. EXPERIMENTAL EVALUTION

Speech recognition experiments have been conducted to evaluate the performance of the new algorithm. The Speaker Independent Mandarin continuous speech recognition system is well trained before adaptation using the database DB863. The testing data are recorded in a stable lab condition. Five male speakers are recorded with no restriction on their speaking style. Three persons are chosen for their accent different from the standard accent.

The main features of our LVSCR are summarized as follows[9]: 12 dimension MFCC, 1 dimension normalized pitch, and their 1 and 2 order derivative MFCC, energy and pitch. The pitch is extracted through the auto-correlation algorithm with the DP algorithm to smooth it. Decision Tree based gender dependent class-triphone modals are trained from the training database DB863. Open LM is trained from the corpus with 387 million words.

For the simplicity of notation, we denote MLLR using a regression class trees as SMLLR. Denote Target-driven MLLR with multiply layer structure as TMLLR.

### 4.1 Comparison the performance of MAP, MLLR SMLLR, and TMMLR.

One speaker M1 is chosen for the first test. The M1's 250 sentences are used to do adaptation. Another 100 sentences are applied to test the performance of the system after adaptation. Variance transform applying a diagonal transformation matrix $\mathbf{H}$ is used in MLLR to give a further improvement after mean adaptation [5]. The original 3000 output distributions are clustered to form 1056 base regression class.

Table 1 Recognition rate after adaptation using the four different adaptation methods

| Baseline | MLLR | MAP | SMLLR | TMLLR |
|---|---|---|---|---|
| 80.2% | 84.1% | 86.3% | 89.4% | 91.6% |

Table 1 gives the recognition accuracy after adaptation using the four different adaptation methods. SMLLR and TMLLR have yielded much better results than the others. TMLLR give a further recognition accuracy improvement by absolute 2.2% over SMLLR.

### 4.2 Compare regression class definition ability of the SMLLR and TMLLR.

SMLLR and TMLLR lie much differences in the their regression class definition. This test checks the relationship between the regression class number and the recognition accuracy after adaptation. The data for adaptation and testing using in this test is the same as that in 4.1. The base regression class number is the same as that in 4.1 too. The recognition accuracy before adaptation is 80.2%.

Table2 the relationship between the recognition accuracy and the regression class number.

| Class Number | SMLLR | TMLLR |
|---|---|---|
| 2 | 84.1% | 84.9% |
| 12 | 87.5% | 88.2% |
| 23 | 88.3% | 89.0% |
| 36 | 88.9% | 89.7% |
| 45 | 89.2% | 90.3% |
| 51 | 89.4% | 90.7% |
| 65 | 89.2% | 91.0% |
| 75 | 88.6% | 91.3% |
| 85 | 86.2% | 91.5% |
| 105 | | 91.6% |
| 120 | | 90.4% |

Table2 give the relationship between the recognition accuracy and the number of the regression class of the two algorithms. SMMLR have the highest accuracy 89.4% when 51 regression classes are defined. After that, more regression classes do no good to the system's accuracy. TMLLR maintain a higher recognition than SMLLR in all regression numbers. It improves its accuracy gradually with the regression class number increase up to the number 105. The highest accuracy of the TMMLR is 91.6%. This test proves that TMLLR have the ability to define more suitable regression classes for the accuracy improvement.

### 4.3 Unsupervised adaptation using TMLLR

Unsupervised adaptation described in [7] is used to justify the new algorithm's ability in defining good regression classes when performing unsupervised adaptation. The original output distributions are clustering to form 201 regression base classes. Fifty sentences Speaker M3's f50 sentences are chosen to do adaptation, and M3's other 100 sentences are used to test the system's performance after adaptation. The recognition accuracy before adaptation is 70.6%.

Table3 Relationship between unsupervised adaptation accuracy and the number of regression classes

| | 2 | 4 | 6 | 8 |
|---|---|---|---|---|
| SMLLR | 74.66% | 72.42% | | |
| TMLLR | 74.66% | 75.21% | 76.10% | 74.36% |

Table3 give the recognition accuracy of SMLLR and TMLLR after unsupervised adaptation. Only 50 sentences of adaptation data are used in this test. From the results listed in the Table4, we can see that SMLLR only take effects using two regression classes, while TMLLR give an significant improvement of the accuracy until 6 regression classes are definition. The final test accuracy of TMLLR is 76.1%, much higher than accuracy of SMLLR by absolute 1.44 percent.

### 4.5 Compare the performance of SMLLR and TMLLR

This test compares the performance of SMLLR and TMLLR using the five speakers' test data. Each speaker's acoustic modal is adapted using his 250 sentences and his another 100 sentence is used to test the system recognition accuracy after adaptation.

Table 4 Recognition accuracy (%) of five speakers before and after adaptation.

| | M1 | M2 | M3 | M4 | M5 | ave |
|---|---|---|---|---|---|---|
| BaseLine | 80.2 | 81.9 | 70.6 | 54.7 | 58.4 | 69.2 |
| SMLLR | 89.4 | 87.6 | 77.7 | 63.1 | 67.8 | 77.1 |
| TMLLR | 91.6 | 86.3 | 83.1 | 64.3 | 72.4 | 79.6 |

Table 4 gives the recognition accuracy of the five male speakers to test the performance of the SMLLR and TMLLR. The average recognition accuracy of the system before adaptation is 69.2%. After adaptation using SMLLR, the average recognition accuracy has been improved to 77.1%. TMLLR give the better recognition accuracy in every test speaker except M2, which is mostly due to the current regression classes number is not fit for this speaker. Large improvement is witnessed in speaker M3 and M5. The possible reason is the two speaker's acoustic property is much different from the speakers in the original acoustic modal training set. For the testing five speakers, TMMLR gives an average recognition accuracy 79.6% and 10% error reduction over SMLLR.

### 4.6 Computational cost of TMLLR

For the experiments in 4.5, it takes 40 minutes to complete a SMLLR adaptation with 250 adaptation sentences (15~25 words) on average, which is running on PC 550 with 128M memory. For TMLLR, only 23 minutes is needed to complete an adaptation process with the same amount adaptation data on the same computer.

## 5. CONCLUSIONS

A novel target-driven MLLR with multiply layer adaptation algorithm is presented in this paper. The algorithm bases on the increment of target function to define the regression classes rather than the prior assumption that those output distributions close together in acoustic should be place in a same regression class. High recognition accuracy after adaptation is obtained in this way. Special calculation structure makes the algorithm have much less calculation complexity. Experimental results show that the new adaptation algorithm has better adaptation result with much less computation time consuming.

## 6. REFERENCE


[1] Gauvain J L, Lee C H. Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains . In: *IEEE Trans. On Speech and Audio Processing ,1994,2(2):291-298.*

[2] Leggetter C J, Woodland P C. Maximum likelihood linear regression for speaker adaptation of continuous density HMMs. In: *Computer Speech and Language. 1995,9:171-186.*

[3] Leggetter C J, Woodland P C, Flexible speaker adaptation using maximum likelihood linear regression. In:*Proceedings Eurospeech. 1995:1155-1158.*

[4] Gales M J F. The Generation and Use of Regression Class Trees For MLLR Adaptation. In: Technical Report CUED/F-INFENG/TR.181. Cambridge University Engineering Department, 1994,June.

[5] Gales M J F. Mean and Variance Adaptation Within the MLLR Framework. In: *Computer Speech and Language. 1996,10: 249-264.*

[6] Young S J, Odell J J, Woodland P C. The Use Of State Tying in Continuous Speech Recognition. In: Proceedings *Eurospeech. 1993*: *2203-2206.*

[7] Tomoko Matsui, Sadaoki Furui. N-Best based unsupervised speaker adaptation for speech recognition. *Computer Speech and Language. 1998 12:41-50.*

[8] Chengrong Li, Jingdong Chen, and Bo Xu. Regression class selection and speaker adaptation with MLLR in mandarin continuous speech recognition. *Eurospeech.1999 : 2503-2506* .

[9] Sheng Gao, Xu Bo, and Taiyi Huang. A new framework for mandarin LVCSR based on one-pass decoder. *Proceedings ISCSLP 2000:49-52.*