Regularized Maximum Likelihood Linear Regression Adaptation for Computer-Assisted Language Learning Systems

Dean LUO††, Yu QIAO††, Nonmembers, Nobuaki MINEMATSU†††, and Keikichi HIROSE†††, Members

SUMMARY This study focuses on speaker adaptation techniques for Computer-Assisted Language Learning (CALL). We first investigate the effects and problems of Maximum Likelihood Linear Regression (MLLR) speaker adaptation when used in pronunciation evaluation. Automatic scoring and error detection experiments are conducted on two publicly available databases of Japanese learners’ English pronunciation. As we expected, over-adaptation causes misjudgment of pronunciation accuracy. Following the analysis, we propose a novel method, Regularized Maximum Likelihood Regression (Regularized-MLLR) adaptation, to solve the problem of the adverse effects of MLLR adaptation. This method uses a group of teachers’ data to regularize learners’ transformation matrices so that erroneous pronunciations will not be erroneously transformed as correct ones. We implement this idea in two ways: one is using the average of the teachers’ transformation matrices as a constraint to MLLR, and the other is using linear combinations of the teachers’ matrices to represent learners’ transformations. Experimental results show that the proposed methods can better utilize MLLR adaptation and avoid over-adaptation.

key words: CALL, speaker adaptation, MLLR, goodness of pronunciation, automatic scoring, pronunciation evaluation, error detection

1. Introduction

One of the biggest challenges in CALL system development is to deal with the mismatches between learners’ speech and the acoustic models. In ASR, speaker adaptation techniques such as maximum likelihood linear regression (MLLR) speaker adaptation [1] have been proved effective in reducing model mismatches. However, instead of recognizing the intended words by the speaker, the purposes of CALL are to evaluate and detect mispronunciations in learners’ speech. When conventional adaptation techniques are directly applied to the acoustic models used in CALL, incorrect pronunciations might be recognized as correct due to over-adaptation. To deal with this problem, [2]–[4] use global MLLR adaptation (with only one and global transformation) to keep the main speaker characteristic while ignoring the pronunciation details. [5] uses a bilingual’s utterances to map learners’ transformation matrices of their native language to the target language. These methods have been proved to outperform baseline systems that are without adaptation. However, to the best of the authors’ knowledge, no quantitative analysis has been reported to investigate the adverse effects of speaker adaptation with learners’ imperfect pronunciations as adaptation data.

This study first investigates the effects of conventional maximum likelihood linear regression (MLLR) speaker adaptation on pronunciation evaluation for CALL in two ways: automatic scoring and phoneme error detection. Following the experimental analyses, we provide solutions to the over adaptation problem. We use a group of teachers’ data to calculate each teacher’s transformation matrix with MLLR, and then use the teachers’ matrices to regularize learners’ transformations. We refer to this method as Regularized-MLLR and implement it in two ways: one is using the average of the teachers’ matrices as a constraint to the conventional MLLR objective function, and the other is using a linear combination of the teachers’ matrices to represent each target learner’s matrix. Experimental results show the high validity of the proposed methods.

2. Pronunciation Evaluation with MLLR Adaptation

MLLR adaptation is an efficient method when the adaptation data is limited. Since comparing with the model parameters, the amount of adaptation data is usually small, these parameters are often clustered into regression classes. By using a regression tree, the number of regression classes can be dynamically determined according to the amount of the data. A set of linear transformations will be applied to all the components that belong to the same regression class [6]. In the case of CALL, the number of nodes of regression trees is usually limited to 1 (which is often called global adaptation) to avoid looking into too many details of learners’ pronunciations [4]. Here, in order to investigate the adverse effects of MLLR on pronunciation evaluation, we examine the changes of performances while increasing the number of nodes of regression trees from 1.

2.1 Acoustic Models

The acoustic models we use for evaluation experiments are triphone HMM models for 41 English phonemes trained on TIMIT [7] and WSJ [8] speech databases with CMU pronunciation dictionary [9]. 39-dimensional feature vectors, consisting of 12-dimensional MFCC, log-energy, and their first and second derivatives, are extracted from utterances.
using a 25 ms-length window shifted every 10 ms. The CMS (Cepstral Mean Subtraction) was applied to each utterance unit. Each HMM has three output states with a left-to-right topology with self-loops and no transitions which skip over states.

2.2 Automatic Scoring

2.2.1 Goodness of Pronunciation

The confidence-based pronunciation assessment, which is defined as the Goodness of Pronunciation (GOP), is often used for assessing learners’ articulation and shows good results [10], [11]. In this study, we use the HMM acoustic models trained on WSJ and TIMIT corpus to calculate GOP scores defined as below. For each acoustic segment \(O^{(p)}\) of phoneme \(p\), \(GOP(O^{(p)})\) is defined as posterior probability by the following log-likelihood ratio.

\[
GOP(O^{(p)}) = \frac{1}{D_p} \log \left( \frac{P(p|O^{(p)})}{\sum_{q \in Q} P(O^{(p)}|q) P(q)} \right)
\]

\[
= \frac{1}{D_p} \log \left( \frac{P(O^{(p)}|p) P(p)}{\sum_{q \in Q} P(O^{(p)}|q) P(q)} \right)
\]

\[
\approx \frac{1}{D_p} \log \left( \frac{P(O^{(p)}|p)}{\max_{q \in Q} P(O^{(p)}|q)} \right)
\]

where \(P(p|O^{(p)})\) is the posterior probability that the speaker uttered phoneme \(p\) given \(O^{(p)}\), \(Q\) is the full set of phonemes, and \(D_p\) is the duration of segment \(O^{(p)}\). The numerator of Eq. 3 can be calculated by scores generated during the forced Viterbi alignment, and the denominator can be approximately attained by continuous phoneme recognition with an unconstrained phone loop grammar.

Since the boundaries of phoneme \(p\) yielded from forced alignment do not necessarily coincide with the boundaries of phoneme \(q\) resulted from continuous phoneme recognition, the frame average log likelihoods of the same speech segment are often used [10].

2.2.2 Database

We use ERJ (English Read by Japanese Students) corpus [12] to calculate GOP scores with MLLR adaptation. In ERJ corpus, for each learner, approximately 120 sentence utterances were recorded. 10 sentences of each learner were chosen and 4 phonetic experts (A to D) who are native speakers of General American English were asked to give a segmental intelligibility score ranging from 1 (poorest) to 5 (best) for each chosen sentence. The average of those scores for each learner is calculated as his or her proficiency score. For our automatic scoring experiments, 42 learners (21 males and 21 females) with higher agreements among raters and a variety of proficiency were selected. The inter-rater correlations are shown in Table 1. Average phoneme GOP score over 30 sentences read by each learner are calculated as an automatic score for the learner. 60 sentence utterances of each learner are used as adaptation data.

2.2.3 Experimental Results

We investigate the correlations between GOP scores and human scores while increasing the number of the nodes of regression trees. Here the number 0 means without adaptation, and 1 represents global adaptation. As shown in Fig. 1, global adaptation yields the best correction of 0.65, yet when the number of nodes of regression class tree increases from 2, the performance drops. When the number is larger than 4, the correlation is even worse than the original models. This clearly indicates when more details of pronunciation are considered during the adaptation, over-adaptation occurs.

2.3 Error Detection

Two most popular methods of error detection are employed for our phoneme error detection experiments: one is based on pronunciation networks [2] and the other is based on GOP scores [10], [11]. The former method predicts possible error patterns and thus is able to detect specified types of errors such as phoneme-level substitution, deletion or insertion. However, the detection performance is largely depending on the size of the pronunciation networks. The latter method often uses a preset threshold to determine whether a phoneme is correctly pronounced or not. Although this method cannot specify the type of an error that occurs, by choosing the optimal threshold for each phoneme, much better detection performance can be obtained.
detection accuracy can be used to measure the performances of pronunciation networks and a larger network often results in lower detection precision. When the number of nodes is larger than 2. This indicates that the number of false rejections drops significantly, thus the precision of the acoustic models with MLLR and Regularized-MLLR.

2.3.1 Database

Because the ERJ database does not contain phoneme labels with erroneous pronunciations, we use another corpus of English words spoken by Japanese students. The database [13] consists of 5,950 utterances of 850 basic English words read by seven Japanese speakers. This database contains manually annotated phonemic labels that were faithfully transcribed and include erroneous phonemes. This database has been used to evaluate the performances of acoustic models for CALL [14]. We used the utterances of 4 speakers (2 males and 2 females) with many typical errors of Japanese learners. For each learner, 450 word utterances are used as adaptation data, and the remaining 400 utterances are used as test data.

2.3.2 Error Detection Based on Network Grammars

The first method we use to detect pronunciation errors is using pronunciation networks that include correct pronunciations and various error patterns to predict learners’ possible mispronunciations. These pronunciation networks are often called network grammars. An example of network grammar is shown in Fig. 2. The network grammar predicts 4 possible errors that might occur when a Japanese learner utter English word “grid”: inserting /uh/ after /g/, substituting /s/ with /l/, substituting /ih/ with /iy/, and inserting /ao/ after /d/ (the phonemic descriptions are based on TIMIT database). Any combination of these 4 possible errors, i.e. 16 alternative paths, can be detected according to the acoustic scores calculated with HMM models. By referring to [15] and [14], 12 major error patterns shown in Table 2 were defined and any irregular errors in the labels were added to the prediction networks. As a result, there are 33 alternative paths per word in average. The error detection performance highly depends on pronunciation networks and a larger network often results in lower detection precision. When the same network is used, the relative increase or decrease of detection accuracy can be used to measure the performances of the acoustic models with MLLR and Regularized-MLLR.

2.3.3 Error Detection Based on GOP Scores

We calculate the phoneme-level GOP scores according to Eq. (3), and use phoneme-dependent thresholds, which are based on mispronunciation labels by experts, to decide if the phonemes are correct or not. We investigate the detection rate on 12 most frequently mispronounced phonemes according to the manual labels. These 12 phonemes are /ih/, /er/, /aw/, /ow/, /ey/, /i/, /j/, /ey/, /l/, /n/, /s/, /l/, /z/ (the phonemic descriptions are based on TIMIT database).

2.3.4 Experimental Results

We use precision and recall rates defined as below to measure the performance of acoustic models with MLLR.

\[
\text{Precision} = \frac{N_{hit}}{N_{total}} = \frac{N_{hit}}{N_{hit} + N_{FR}} \quad (4)
\]

\[
\text{Recall} = \frac{N_{hit}}{N_{labeled}} \quad (5)
\]

where \(N_{hit}\) represents the number of the errors that were correctly detected, \(N_{total}\) is the total number of detected errors, \(N_{FR}\) is the number of false rejections and \(N_{labeled}\) is the number of all the errors that were detected by phoneticians, and F-measure defined as below is also calculated to combine the two measures.

\[
F\text{-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (6)
\]

Figure 3 shows the performances of error detection based on network grammar with MLLR adaptation. Although the precision rates keep increasing when more transforms are used for adaptation, the recall rate drops when the number of nodes is larger than 2. This indicates that with adaptation to reduce model mismatches, the number of false rejections \(N_{FR}\) drops significantly, thus the precision rate increases. However, since the number \(N_{labeled}\) is only

<table>
<thead>
<tr>
<th>Table 2</th>
<th>12 basic error patterns for constructing network grammars.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error pattern</td>
<td>Example</td>
</tr>
<tr>
<td>er/ah substitution</td>
<td>paper</td>
</tr>
<tr>
<td>ih/iy substitution</td>
<td>little</td>
</tr>
<tr>
<td>v/b substitution</td>
<td>very</td>
</tr>
<tr>
<td>s/sh substitution</td>
<td>sea</td>
</tr>
<tr>
<td>ch/sh substitution</td>
<td>choose</td>
</tr>
<tr>
<td>w/y deletion</td>
<td>would</td>
</tr>
<tr>
<td>r/l substitution</td>
<td>road</td>
</tr>
<tr>
<td>Word-final vowel insertion</td>
<td>let</td>
</tr>
<tr>
<td>Vowel non-reduction</td>
<td>student</td>
</tr>
<tr>
<td>VCC-cluster vowel insertion</td>
<td>active</td>
</tr>
<tr>
<td>CCC-ccuster vowel insertion</td>
<td>study</td>
</tr>
<tr>
<td>f/h substitution</td>
<td>fire</td>
</tr>
</tbody>
</table>
decided by the label, the decrease of recall means the
decrease of the number of correctly detected errors \( N_{\text{hit}} \). This
result shows that over-adaptation can cause more errors to
be recognized as correct pronunciations (i.e. the number of
false acceptances increases), yet at the same time, even with
over-adaptation, more false rejections can be reduced and as
a result, the precision rates keep increasing.

For the error detection method based on GOP scores,
the recall and precision can be adjusted by changing the
values of the thresholds. According to [11], erroneously re-
jecting correct pronunciations would be more detrimental
for learners than erroneously accepting mispronunciations.
Thus we need to keep the false rejection rate at relatively low
level, which means to keep the precision relatively high, and
find the optimal thresholds that maximize the recall. Here,
we investigate the change of recalls at precision level of
70% while increasing the number of regression classes for
MLLR. Again, the number 0 means no adaptation, i.e. using
the original acoustic models. As shown in Fig. 4, only global
adaptation shows slight improvement over original models
and when the number of regression classes is larger that 2,
the performance drops significantly. This clearly indicates
that over adaptation occurs with MLLR.

2.4 MLLR Adaptation with Phone Recognizer

Instead of using phoneme labels attained by using trans-
script and a pronunciation dictionary for adaptation, we can
use phoneme labels generated through a phone reconizer
based on network grammar to reduce erroneous phonemes
in adaptation data. We refer to this approach as MLLR2.
The comparison results of MLLR2 and MLLR are shown
in Fig. 5, Fig. 6 and Fig. 7. Figure 5 shows that MLLR2
cannot outperform MLLR global adaptation, although over-
adaptation problem is slightly improved when the number
of regression classes increases from 1. In Fig. 6, although
MLLR2 improves recall rates slightly, precision rates are

Fig. 3 Error detection performances based on network grammar.

Fig. 4 Error detection performances based on GOP scores.

Fig. 5 Automatic scoring with MLLR2 and MLLR.

Fig. 6 Error detection performances based on network grammar with
MLLR2 and MLLR.
worse than MLLR. Figure 7 also indicate that only when the number of regression classes is larger than 2, MLLR2 outperforms MLLR. We consider that this is because the phoneme recognition results are not accurate enough for these tasks. We need to find new methods that can at least outperform MLLR global adaptation and preferably prevent over-adaptation problem.

3. Regularized-MLLR

The results of automatic scoring and error detection experiments clearly show the adverse and good effects of MLLR adaptation on pronunciation evaluation. If we can solve the problem of “bad judgment” of adapted models, we might be able to achieve both better recall and precision. Regularized-MLLR is one possible solution to this problem.

3.1 The First Implementation of Regularized-MLLR

Our first implementation of Regularized-MLLR is using the average of a group of teachers’ matrices as a constraint to conventional MLLR.

The standard auxiliary function for MLLR is defined as below to estimate the transform \(W_r\) for the \(M_r\) Gaussian components \(\{m_1, m_2, \ldots, m_{M_r}\}\) that are tied together in each regression class \(r\), as determined by the regression class tree.

\[
Q(M, \hat{M}) = \frac{1}{2} \sum_{r=1}^{R} \sum_{m=1}^{M_r} \sum_{t=1}^{T} L_{m_r}(t) \times
\left[ k^{(m)} + \log(|\Sigma_m|) + (\alpha(t) - \hat{\mu}_m)^T \Sigma_m^{-1}(\alpha(t) - \hat{\mu}_m) \right]
\]

where \(M\) is the original HMM model set, \(M\) is the adapted model set, \(R\) is the number of the nodes of regression tree, \(M_r\) is the number of Gaussian components that are to be tied together, \(k^{(m)}\) subsumes all constants, and \(\hat{\mu}_m\) and \(\Sigma_m\) are the adapted mean vector and covariance matrix for the mixture component \(m_r\) respectively, and \(L_{m_r}(t)\) is the occupation likelihood defined as

\[
L_{m_r}(t) = p(q_{m_r}(t) | M, O_T),
\]

and \(q_{m_r}(t)\) indicates the Gaussian component \(m_r\) at time \(t\), and \(O_T = \{\alpha(1), \ldots, \alpha(T)\}\) is the adaptation data. Let \(\{W_r^C, \ldots, W_r^{C_n}\}\) denote a set of transformation matrices estimated from a group of \(N\) teachers’ data. \(W_r^C = \frac{1}{N} \sum_n W_r^C\) represents the mean of these matrices. The first objective function for Regularized-MLLR is defined as:

\[
\max Q(M, \hat{M}) - \lambda \|W_r - W_r^C\|_F^2.
\]

where \(\lambda\) is a parameter depending on the acoustic characteristics of the speaker. In traditional MLLR, \(W_r\) is calculated by maximizing \(Q(M, \hat{M})\). In the proposed method, however, over-adaptation is avoided by the 2nd term of Eq. 9. This term functions as penalty of changing the model parameters so radically. We assume diagonal covariance matrices and the adaptation is only applied to the mean vector for each Gaussian component,

\[
diag(\Sigma_m) = [\sigma_{m_1}^2, \sigma_{m_2}^2, \ldots, \sigma_{m,n}^2],
\]

\[
\Sigma_m = \Sigma_m^r,
\]

\[
\hat{\mu}_m = W_r \xi_m^r,
\]

where \(\xi_m^r\) is the extended mean vector for the Gaussian component \(m_r\) defined as,

\[
\xi_m = [1 \mu_1 \mu_2 \ldots \mu_d]^T,
\]

where \(d\) is the dimensionality of the data. Considering the row decomposition \(W_r = [w_{r,1}; w_{r,2}; \ldots; w_{r,d}]\), we can calculate the cost function for each row vector,

\[
f(w_{r,j}) = \sum_{m=1}^{M_r} \sum_{t=1}^{T} L_{m_r}(t) \frac{1}{\sigma_{m,j}^2} (\alpha(t) - w_{r,j} \xi_m^r)^2
\]

\[
-\lambda(w_{r,j} - w_{r,j}^C)(w_{r,j} - w_{r,j}^C)^T
\]

\[
= K_j + w_{r,j} \sum_{m=1}^{M_r} \sum_{t=1}^{T} L_{m_r}(t) \frac{1}{\sigma_{m,j}^2} \xi_m^T \xi_m - \lambda \xi_m^T \xi_m - 2w_{r,j} \sum_{m=1}^{M_r} \sum_{t=1}^{T} L_{m_r}(t) \frac{1}{\sigma_{m,j}^2} \alpha(t) \xi_m^T - \lambda \xi_m^T \xi_m - \lambda \xi_m^T \alpha(T),
\]

where \(K_j\) is a constant which does not depend on \(w_{r,j}\). If we set

\[
H_{j}^r = \sum_{m=1}^{M_r} \frac{1}{\sigma_{m,j}^2} \xi_m^T \xi_m - \lambda \xi_m^T \xi_m + \sum_{t=1}^{T} L_{m_r}(t) - \lambda \alpha(T),
\]

\[
N_{j}^r = \sum_{m=1}^{M_r} \sum_{t=1}^{T} L_{m_r}(t) \frac{1}{\sigma_{m,j}^2} \alpha(t) \xi_m^T - \lambda \xi_m^T \alpha(T),
\]

the optimal \(w_{r,j}\) is given by solving

\[
\frac{\partial f(w_{r,j})}{\partial w_{r,j}} = 0,
\]

which yields,

\[
\hat{w}_{r,j} = N_{j}^{rH_{j}^{-1}}.
\]

We will refer to this implementation of Regularized-MLLR as R-MLLR1 hereafter.
3.2 The 2nd Implementation of Regularized-MLLR

R-MLLR1 uses the average of a group of teachers’ transformations as a constraint added to conventional MLLR. The scale of that constraint, which is decided by the parameter \( \lambda \), needs to be manually adjusted for each learner according to his/her acoustic characteristics. When the number of learners is very large, it can be very time-consuming to find an optimal parameter for each learner. Therefore we try another approach which can automatically estimate optimal parameters for different learners.

In the 2nd regularization, we assume a learner’s transformation matrix \( W_r \) can be written as a linear combination of a group of \( N \) teachers’ transformation matrices \( \{ W_r^{C_1}, \ldots, W_r^{C_N} \} \):

\[
W_r = \sum_{n=1}^{N} \alpha_n W_r^{C_n}.
\]

Then the objective function becomes,

\[
\max_{\{\alpha_n\}} g(\alpha_1, \alpha_2, \ldots, \alpha_N) = \sum_{m_r=1}^{M_r} \sum_{r=1}^{T} L_{m_r}(t) \times (\alpha(t) - \sum_{n=1}^{N} \alpha_n W_r^{C_n} \xi_m) T \sum_{m_r=1}^{M_r} (\alpha(t) - \sum_{n=1}^{N} \alpha_n W_r^{C_n} \xi_m) = 0
\]

(22)

and changing \( j = 1, 2, \ldots, N \), we have \( N \) linear equations on \( \{\alpha_n\} \). Then, we can calculate the optimal \( \{\alpha_n\} \) by solving these linear equations. For simplicity, we can set

\[
\xi_{m_r,j} = W_r^{C_j} \xi_m.
\]

The linear equations become,

\[
\sum_{m_r=1}^{M_r} \sum_{r=1}^{T} L_{m_r}(t) \sum_{m_r=1}^{M_r} (\alpha(t) - \sum_{n=1}^{N} \alpha_n \xi_{m_r,n}) \xi_{m_r,j} = 0
\]

(24)

We will refer to this implementation of Regularized-MLLR as R-MLLR2 hereafter. In R-MLLR2, by introducing the teachers’ transformations, we are using learners’ speech data to find out which combination of teachers is most similar to a specific learner in terms of speaker characteristics. Since we do not use learners’ transforms estimated directly from their imperfect pronunciations with MLLR, radical transformations can be avoided.

4. Evaluation Experiments with Regularized-MLLR

In order to prove the validity of our proposed methods, we directly compare the effects of R-MLLR1, R-MLLR2 and MLLR on automatic scoring and error detection. The databases and experiment conditions are the same as Sect. 2. To regularize MLLR transformation, we use 20 teachers’ utterances from ERJ database. These teachers are native speakers of General American English. 60 sentence utterances of each teacher are used to calculate his or her transformation matrices.

4.1 Automatic Scoring Results

We apply transformations estimated with R-MLLR1 and R-MLLR2 to original HMM models by increasing the nodes of regression trees from 1 to 64. And then we use the adapted models to calculate average phoneme GOP score for each learner. The correlations between manual scores and GOP scores with adapted models are shown in Fig. 8. The learners and the amount of evaluation data and adaptation data are the same as Sect. 2.3.2.

As shown in Fig. 8, R-MLLR1 and R-MLLR2 show better performance than conventional MLLR. When the number of regression classes increases after 1(global adaptation), the effect of regularization becomes rather obvious. Although by adding some amount of constraints, R-MLLR1 reduces the adverse effects of over-adaptation, the performance still drops when the number of classes is larger than 1. In the case of R-MLLR2, however, it not only always shows the best results, the performance never drops. This can be explained that in the case of R-MLLR2, the direct use of learners’ transformations estimated by their imperfect pronunciations with MLLR is avoided. However, in the case of R-MLLR1, these transformations are still used and since there is not sufficient labeled data for each learner, the
constraint scale parameter \( \lambda \) manually chosen for each of the 42 learners might not be optimal to yield best results.

4.2 Error Detection Results

We apply transformations estimated with R-MLLR1 and R-MLLR2 to original HMM models by increasing the nodes of regression trees from 1 to 16. And then we use the adapted models to perform error detection experiments with network grammar and also with GOP scores. Experimental setups are the same as error detection experiments with MLLR adaptation. For regularization, the same 20 teachers' utterances from the ERJ database are used as automatic scoring experiments with R-MLLR1 and R-MLLR2.

4.2.1 Error Detection Based on Network Grammar

The results of error detection based on network grammar by comparing any two of the 3 adaptation methods are shown in Fig. 9, Fig. 10, and Fig. 11. As shown in Fig. 9 and Fig. 10, R-MLLR1 and R-MLLR2 improve recall significantly and also keep very high precision rates. Especially in the case of R-MLLR2, recall keeps high level when the number of classes increases. This indicates the proposed method not only benefits from reduction of mismatches (increase of precision) but also prevents over-adaptation. The comparison of R-MLLR1 and R-MLLR2 is shown in Fig. 11. When the number of regression classes is larger than 4, in the case of R-MLLR1, the performance of recall drops, which indicates the over-adaptation problem still occurs. This problem is solved by R-MLLR2, and the performances of precision of R-MLLR1 and R-MLLR2 are almost the same.

4.2.2 Error Detection Based on GOP Scores

Figure 12 shows the performances of recall at the precision level of 70%. R-MLLR1 improves the performances comparing with MLLR, yet over-adaptation still remains. R-MLLR1 and R-MLLR2 are almost the same.

Fig. 9 Error detection results based on network grammar, comparing R-MLLR1 with MLLR.

Fig. 10 Error detection results based on network grammar, comparing R-MLLR2 with MLLR.

Fig. 11 Error detection results based on network grammar, comparing R-MLLR1 with R-MLLR2.

Fig. 12 Recall of error detection results based on GOP scores at the precision level of 70%.
MLLR2 outperforms MLLR or R-MLLR1, especially when the number of regression classes becomes larger. This again shows that this method can avoid the over-adaptation problems by using linear combination of teachers’ MLLR transformations instead of their own transformation matrices.

5. Conclusions and Future Work

5.1 Conclusions

In this paper, we propose a novel adaptation technique, Regularized Maximum Likelihood Linear Regression (Regularized-MLLR), for CALL systems. The idea is to use a group of teachers’ transformations to regularize learners’ transformations so that erroneous pronunciations will not be transformed into good pronunciations.

First, we investigate the effects of MLLR on pronunciation evaluation in two ways: automatic scoring and error detection. Experimental results show that although the MLLR global adaptation (the number of regression classes is 1) can indeed improve evaluation performances, when the number of regression classes increases and more details of learners’ pronunciations are adapted, over-adaptation occurs so that erroneous pronunciations are recognized as correct ones. However, even with over-adaptation, conventional adaptation can still improve precision rate of error detection performance, which indicates that false rejections can be reduced by conventional MLLR.

Following these results, we implement two forms of Regularized-MLLR, R-MLLR1 and R-MLLR2 by using teachers’ perfect pronunciations to regularize learners’ transformations. R-MLLR1 uses the average of a group of teachers’ transformation matrices as a constraint added to the conventional MLLR transformations. This constraint prevents radical transformations when there are too many errors in the adaptation data. R-MLLR2 uses linear combination of the teachers’ MLLR transformation matrices to represent each learner’s transformation. This approach does not directly use learners’ MLLR transformations that are estimated from their imperfect pronunciations, therefore prevents over-adaptation.

We compare R-MLLR1 and R-MLLR2 with conventional MLLR by conducting experiments under the same conditions used to investigate the adverse effects of MLLR. Automatic scoring and error detection experiments show that the proposed methods outperform conventional MLLR. By adding constraints to MLLR, R-MLLR1 indeed reduces the adverse effects of MLLR, yet performances still drop due to over-adaptation. R-MLLR2 not only out-performs MLLR global adaptation, which is widely use for CALL, but also prevents over-adaptation by using linear combinations of teachers’ matrices instead of using learners’ directly. The proposed methods can better utilize speaker adaptation and prevent the adverse effects, thus more suitable for CALL systems.

5.2 Future Work

To regularize learners’ transformation, we only use the 20 teachers’ speech data from ERJ database. Increasing the number of teachers increases the variety of speaker characteristics of the teachers’ data we use for regulation. We need to investigate if increasing the number of teachers would improve the effectiveness of adaptation.

The method we use to cluster model parameters into regression classes for MLLR and Regularized-MLLR is according to how close they are in acoustic space. By using some phonetic expertise in deciding which components should be clustered together, we might obtain better recognition and error detection results.

We use publicly available databases for evaluation experiments, and they are not specially designed for our task. We are planning to develop databases that can allow us to closely examine the relationship between adaptation data and the performances of pronunciation evaluation.

Other adaptation techniques such as MAP and Eigen-voices need to be examined and compared with MLLR-based methods.

Since the proposed methods are language-independent for pronunciation evaluation, we would also like to test them on different databases of different languages.

Acknowledgments

The authors would like to thank Dr. Yasushi Tsubota and Prof. Masatake Dantsuji at Kyoto University for providing the database of Basic English Words Read by Japanese Students, and Prof. Yutaka Yamauchi at Tokyo International University for educational insights.

References


Dean Luo received his B.E degree from Tokyo Institute of Technology and M.E. degree from the University of Tokyo in 2005 and 2007 respectively. Currently, he is a graduate student of Graduate School of Engineering, the University of Tokyo. His research interests include speech recognition, computer-assisted language learning. He is a student member of IEEE, ISCA and ASJ.

Yu Qiao received his B.E. degree and M.E. degree from Chongqing University, China, in 2000 and 2003, respectively. He received his Ph.D from the University of Electro-Communications, Japan, in 2006. He was a postdoctoral researcher with the Chinese Univ. of Hong Kong from 2006 to 2007. Then he became a JSPS fellow, a researcher and a project assistant professor with the University of Tokyo. Now he is an associated professor, the executive director of multimedia laboratory with the Shenzhen Institutes of Advanced Technology, the Chinese Academy of Science. His research interests include speech processing, pattern recognition, computer vision, multimedia, handwriting analysis and machine learning. He has published more than 70 papers in these fields. He is a member of IEEE and ISCA.

Nobuaki Minematsu received the Ph.D. degree in electronic engineering in 1995 from The University of Tokyo. Currently, he is an associate professor of Graduate School of Information Science and Technology, The University of Tokyo. He has a very wide interest in speech communication from science to engineering, including linguistics, psychology, language disorders, and musicology. He is a member of IEEE, ISCA, IPA, the Computer Assisted Language Instruction Consortium, the Institute of Electronics, Information and Communication Engineers, the Acoustical Society of Japan, Information Processing Society of Japan, Japanese Society for Artificial Intelligence, and Phonetic Society of Japan.

Keikichi Hirose received the B.E. and Ph.D. degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1972 and in 1977 respectively. Currently, he is a professor of Graduate School of Information Science and Technology, The University of Tokyo. His research interests widely cover spoken language information processing, especially the topics related to prosody: analysis prosodic features using F0 contour generation process model, generation of F0 contours for various styles of speech synthesis, use of prosodic features for speech recognition, concept-to-speech conversion for spoken dialogue systems, pronunciation training systems for accents and stresses, and so on. He is a member of Permanent Council of International Conference on Spoken Language processing (ICSLP), a member of Advisory Committee of ISCA, an Editorial committee member of ISCA, an Advisory Committee member of the Speech Committee at IEICE/ASJ, and vice president of Signal Processing Society of Japan.