Supervised Detection and Unsupervised Discovery of Pronunciation Error Patterns for Computer-Assisted Language Learning

Yow-Bang Wang and Lin-shan Lee, Fellow, IEEE

Abstract—Pronunciation error patterns (EPs) are patterns of mispronunciation frequently produced by language learners, and are usually different for different pairs of target and native languages. Accurate information of EPs can offer helpful feedbacks to the learners to improve their language skills. However, the major difficulty of EP detection comes from the fact that EPs are intrinsically similar to their corresponding canonical pronunciation, and different EPs corresponding to same canonical pronunciation are also intrinsically similar to each other. As a result, distinguishing EPs from their corresponding canonical pronunciation and between different EPs of the same phoneme is a difficult task—perhaps even more difficult than distinguishing between different phonemes in one language. On the other hand, the cost of deriving all EPs for each pair of target and native languages is high, usually requiring extensive expert knowledge or high-quality annotated data. Unsupervised EP discovery from a corpus of learner recordings would thus be an attractive addition to the field.

In this paper, we propose new frameworks for both supervised EP detection and unsupervised EP discovery. For supervised EP detection, we use hierarchical multi-layer perceptrons (MLPs) as the EP classifier, trained with the help of GMM/MECM using HMM/GMM in a two-pass Viterbi decoding architecture. Experimental results show that the new framework enhances the power of EP diagnosis. For unsupervised EP discovery we propose the first known framework, using the hierarchical agglomerative clustering (HAC) algorithm to explore sub-segmental variation within phoneme segments and produce fixed-length segment-level feature vectors in order to distinguish different EPs. We tested K-means (assuming a known number of EPs) and the Gaussian mixture model with the minimum description length principle (estimating an unknown number of EPs) for EP discovery. Preliminary experiments offered very encouraging results, although there is still a long way to go to approach the performance of human experts. We also propose to use the universal phoneme posterioriogram (UPP), derived from an MLP trained on corpora of mixed languages, as frame-level features in both supervised detection and unsupervised discovery of EPs. Experimental results show that using UPP not only achieves the best performance, but also is useful in analyzing the mispronunciation produced by language learners.

Index Terms—Computer-Assisted Language Learning, Computer-Aided Pronunciation Training, Error Pattern Detection, Error Pattern Discovery, Universal Phoneme Posterioriogram

I. INTRODUCTION

In this globalized era, it is generally recognized that speaking two or more languages is not only advantageous but in fact necessary. Also, the application of speech processing technology in second language learning has drawn much attention in recent years, with more and more related products entering the market. These trends have led to substantial efforts toward Computer-assisted language learning (CALL) to meet the strong demand of second language learning. Below are several examples from research projects, groups, and achievements along this direction: Granström [1] presented “Virtual Language Tutor” for Swedish learning, with a 3D talking head as a virtual tutor that interacts with the learners [2]; Tsurutani et. al. [3] developed a program for self-assessment of Japanese pronunciation by English-speaking learners; Abdou et. al. [4] describe the HAFSS system, developed to teach Arabic pronunciations to non-native speakers; Alwan et. al. [5] reported on the system developed in the TBALL project aimed at assessing the English literacy skills of children in grades K–2; Strik et. al. [6] developed a web-based role-playing environment for practicing conversation in Dutch for the DISCO project; Zechner et. al. [7] presented the SpeechRater as an online practice test for the Test of English as a Foreign Language internet-based test (TOEFL iBT); Yoshimoto [8] developed the web-based “Rainbow Rummy” game for vocabulary acquisition; Duchateau et. al. [9] reported on efforts to assess the Dutch reading level of children in the SPACE project; Wang et. al. [10] reported on the system CALLJ, which prompts students to create sentences to describe certain concepts to help them learn Japanese as a second language; last but not least, Harrison et. al. [11] developed a prototype of the CHELSEA system for Chinese learners of English.

CALL systems can be applied in many different ways, depending on the target learner group and the pedagogical purpose. Computer-aided pronunciation training (CAPT), for instance, analyzes the produced utterance to offer feedback to the language learner in the form of quantitative or qualitative evaluations of the pronunciation proficiency. This process is also referred to as pronunciation evaluation. Quite several well-known pronunciation evaluation metrics are based on the posterior probabilities derived with automatic speech recognition (ASR) technologies. One such example is the goodness-of-pronunciation (GOP) [12], the posterior probability that the speaker uttered a certain phoneme given the corresponding acoustic segment. Substantial research has been conducted on improving these posterior probability-based scores [13][14].

However, in order to generate more informative feedback so learners can improve their pronunciation, people also sought to offer not only quantitative measures of language proficiency, but also specific types of errors the learners have made.
Such error types are usually referred to as error patterns (EPs). In general, EPs are patterns of erroneous pronunciations frequently produced by language learners, usually caused by articulator mechanisms or acoustic phenomena present in the target language but missing in the native language of the learners. EPs are therefore different for different pairs of target and native languages, and can be very complicated for certain target languages when learners speak a wide variety of native languages.

In general, there are at least two different but closely related EP tasks in CAPT: First, to derive the EP dictionary for a given L1-L2 pair, or a given L2 but non-specific L1; second, to verify whether a voice segment produced by a learner is correct, or if it belongs to a specific EP based on the EP dictionary. In this paper the former task is referred to as EP derivation or EP discovery, the latter as EP detection.

The most direct approach to deriving the dictionary of EPs is to begin with the literature of second language learning for a given L1-L2 pair [3][15][16]. Another popular approach is to compare the orthographic transcriptions of a corpus to the actual pronunciations annotated by human listeners [17], or to the free-phone recognition output from an ASR engine [18][19]. The latter approach of using free-phone ASR output can potentially handle a non-specific L1 for a given L2, but tends to depend heavily on ASR performance and on the quality of the data.

Note that the above approaches for EP derivation all require either expert knowledge, time-consuming manual labeling, or reliable ASR results, all of which are expensive. On the other hand, substantial effort has been made in recent years on unsupervised acoustic pattern discovery [20][21][22], with a wide variety of goals including speech recognition, spoken term detection, and out-of-vocabulary (OOV) word modeling. In these efforts the human-annotated data for acoustic model (AM) training in speech recognition is no longer needed. The goal of these works is in general to automatically discover the acoustic patterns in a data set based on the signal characteristics. One of the motivations of such works is that it can be costly to produce carefully annotated data sets for training AM and building ASR system with the traditional hidden Markov model (HMM) framework. Similar to the EP derivation task considered here, well-annotated corpora of EPs for many different application conditions is also very difficult to collect. In fact, for EP derivation the required level of expertise to define and label EPs may be even higher, more difficult to obtain, and more expensive. This is an important motivation to incorporate the concept of unsupervised acoustic pattern discovery into EP derivation.

Given the EP dictionary, the next step is to detect the EPs in the utterances of language learners. A popular approach is to consider the EPs as the pronunciation variations of their corresponding canonical pronunciation. By expanding the recognition network with the EP dictionary and rules, the decoder can automatically find the most probable surface pronunciation of a sentence, which may include one or more EPs [9][23].

The major difficulty of EP detection comes from the fact that EPs are intrinsically similar to their corresponding canonical pronunciation, and that EPs corresponding to the same canonical pronunciation are also intrinsically similar to each other. As a result, distinguishing EPs from their corresponding canonical pronunciation and between different EPs of the same phoneme is a difficult task – perhaps even more difficult than distinguishing between different phonemes in one language. In searching for better discriminability for such a difficult task, more powerful classifiers have been used, often with log-likelihood ratios or posterior probability vectors as input features [14][24][25].

In this paper, we propose new frameworks for both supervised EP detection and unsupervised EP discovery, and report the results tested on a corpus from Mandarin Chinese learners with a wide range of native languages. We propose to use heterogeneous model initialization and concatenated model adaptation techniques to create AM for each EP; and we combine the judgments from these EP models with additional EP classifiers built with hierarchical multi-layer perceptrons (MLPs) for better EP detection. We further propose a novel framework towards unsupervised discovery of EPs from a corpus of learners’ recordings without relying on expert knowledge. We utilize hierarchical agglomerative clustering (HAC) to construct segment-level feature vectors from frame-level feature vectors in this framework. Also, for both the tasks of EP derivation and EP detection, we propose to use the universal phoneme posteriorgram (UPP)[21] as a more discriminative and robust set of frame-level features in distinguishing and analyzing the EPs. These UPP features are posterior probabilities for all possible phonemes across different languages given each signal frame, obtained from neural networks trained with a mixed-language corpus.

The rest of this paper is organized as follows. Section II explains the concept of pronunciation error pattern, the constraints and challenges were facing and how they relate to our approaches. Section III describes the data set in this work, including the EP definition and labeling. Section IV explains the concept of universal phoneme posteriorgram (UPP) and why it can help in distinguishing EPs. Sections V and VI then present respectively the complete frameworks for supervised detection and unsupervised discovery of EPs, and discuss in detail the experimental results. Concluding remarks are finally made in the last section.

II. PRONUNCIATION ERROR PATTERNS AND BASIC RATIONALE OF THIS WORK

We illustrate the basic concept of EPs with Fig. 1. In Fig. 1, we assume the instances of a certain phoneme (e.g., Chinese /l/) produced by different speakers are scattered over a conceptual pronunciation space (e.g. spanned by MFCCs or similar, as shown). The red circle stands for the range of acceptable correct pronunciation, or those produced by native speakers. So the pronunciation instances inside this range are judged correct; those outside are judged incorrect. Among those mispronounced instances, some are relatively close to a certain phoneme in the same or a different language, for example, Chinese /zl/ and English /l/ as shown in Fig. 1, perhaps due to the influence of the language learners’ native languages. These instances are thus clustered and summarized...
as the EPs by the human experts, as was done for the corpus we used here. For example, in Fig. 1 those mispronounced instances close to, or within, the range of the canonical pronunciation of English /l/, are labeled as the error pattern “\(l_{010}\)” by the human experts, meaning “pronounced similar to English /\(l/\)”; and those close to or within the range of the canonical pronunciation of Chinese \(/\l/\) are labeled as error pattern “\(l_{020}\)”, meaning “pronounced similar to Chinese /\(l/\)”. From this figure one may also realize that, as we stated in the introduction, the EPs are intrinsically similar to their corresponding canonical pronunciation and similar to each other as well. As a consequence, detecting EPs (e.g. between Chinese \(/\l/\) and “\(l_{010}\)” and “\(l_{020}\)”) can be more difficult than distinguishing different phonemes in one single language (e.g. between Chinese \(/\l/\) and Chinese \(/\r/\)). Similarly, deriving or discovering EPs can be very difficult too.

Due to the high difficulty of this problem, and the limited size of the data set (produced by language learners and annotated by language teachers) available in this work as will be presented below in Section III, we developed the basic rationale behind the various approaches we propose throughout this paper:

1) First, we aimed at properly utilizing other much larger multi-speaker or even mixing-language corpora, not limited to those produced by language learners. In Section IV, such larger corpora were used for deriving the universal phoneme posteriorgram (UPP) to be used in both supervised EP detection and unsupervised EP discovery. Also, in Section V, for supervised EP detection we generated the EP acoustic models based on the existing phoneme models trained with other larger corpora.

2) Second, knowing the pronunciation EPs are very subtle, or even partial, deviation from their corresponding canonical pronunciation, we made our best effort to boost the discriminability and granularity in our system. In Section IV we derived the UPP with an MLP in order to generate more discriminative frame-level features. In Section V for supervised EP detection, we also adopted hierarchical MLPs for better capability of EP classification. And in Section VI for unsupervised EP discovery, we further construct segment-level feature with hierarchical agglomerative clustering (HAC) to better retain the sub-segmental differences among EPs with variable analysis granularity.

III. DATA SETS FROM THE NTU CHINESE PROJECT

A. Data collection

As described in previous works [26][25][27], we have been collaborating with the Chinese language teachers from the International Chinese Language Program (ICLP) of National Taiwan University (NTU), working on computer-aided pronunciation training for Mandarin Chinese learning. The corpus used here was collected in the years 2008 and 2009. A total of 278 ICLP learners from 36 different countries with balanced gender and a wide variety of native languages were recorded. Each learner was asked to produce 30 sentences, each of which contained 6 to 24 characters. The recording text prompts were chosen so as to cover as many Chinese syllables and tone patterns as possible, and were selected from the learning materials designed by ICLP language teachers and used in NTU Chinese [28], a successful online Chinese pronunciation learning application. We divided the corpus into adaptation, development, and testing sets as shown in Table I.

Note that the percentage of mispronounced segments in each data set is relatively small (around 10%), obviously because the learners from whom we collected these data already had some basic training in Mandarin Chinese pronunciation. This may seems to imply the data set is not too difficult to work with, since for each utterance there are roughly 90% segments that are correctly pronounced. However, one must note that our task is to detect or discover the EPs from the roughly 10% of mispronunciations out of the 90% of correct pronunciations. On the other hand, since the EPs were defined by language teachers based on their linguistic knowledge and pedagogical experience, not based on the corpus we collected, the number of error patterns we are supposed to detect, or to discover, is in fact independent of the data set we used. As a result, no matter how few the instances of a certain EP in our corpus is, that EP is always a target for detection or discovery. The very limited number of instances for each EP thus in fact led to difficulties in this task.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Total Length (hr)</th>
<th>Percentage of mispronunciation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptation</td>
<td>8.4</td>
<td>10.12%</td>
</tr>
<tr>
<td>Development</td>
<td>2.2</td>
<td>8.55%</td>
</tr>
<tr>
<td>Testing</td>
<td>2.0</td>
<td>9.30%</td>
</tr>
</tbody>
</table>

B. Error Pattern definition and labeling

The reference EPs used in this work were summarized by the language teachers of ICLP of NTU based on their
linguistic knowledge and pedagogical experience. These EPs are intended to cover the most frequent EPs made by Mandarin Chinese learners, not limited to any specific corpus, including the NTU Chinese corpus used here. The basic unit used in our EP definition is the Mandarin phoneme set represented in Zhuyin, which is comprised of 39 canonical Mandarin phoneme units. Tables II(a) and II(b) are two examples of our EP definition for a consonant and a diphthong respectively. In the left “ID” columns of these tables, the canonical pronunciations are coded “000”. The code “099” means “none of the above”, i.e., the acoustic segment is neither pronounced correctly, nor can be categorized as any of the EPs. Except for “000” and “099”, the codes denote the EPs for each phoneme. In the right “description” column, “CH” and “EN” respectively indicate the Chinese or English phonemes used to describe the EP. Note that the EP definition includes not only phoneme-level substitutions (e.g., l≈3 in Table II(a)), but also insertions (ei_010 in Table II(b)) and deletions (ei_020 in Table II(b)). The digits in the ID codes are for annotation only – they do not indicate the number of EPs corresponding to each phoneme.

Note that while we describe the pronunciation of the EP “l≈3” as “similar to English /l/”, the acoustic realization of the EP may not be located exactly where the annotated phoneme (English /l/) is, since it simply serves as the closest representation of the EP in the acoustic space, as depicted in Fig. 1; this applies to other EPs as well. Also, although these EPs are described with phonemes from a certain language, these EPs and the descriptions have nothing to do with the learner’s L1. Learners whose native languages are not English may still utter EPs defined with English phonemes.

Two annotators labeled the surface pronunciation of each acoustic segment in each utterance in the above corpus as correct pronunciation or one of the EPs. Ideally, the evaluation reported below should be based on the average of those obtained with the two sets of labels given by the two human annotators. However, among the two annotators who labeled the surface pronunciation of each acoustic segment in learners recordings, only one labeled the whole corpus, while the other labeled only a part of the corpus. Therefore all we can do is to use the labels from the first annotator as the reference patterns in our experiments, and the second one for estimating the consistency between human annotators. Although such results seem incomplete, this is the best we can do with the very limited annotated data.

### IV. Universal Phoneme Posteriorgram (UPP)

In this work, the universal phoneme posteriorgram (UPP) is used as the fundamental frame-level features for the tasks of supervised EP detection and unsupervised EP discovery. The posterior probability has been widely used in CAPT and unsupervised acoustic pattern discovery as well as many other areas. The well-known GOP is calculated based on the posterior probability of the target pronunciation [12]. Some work uses the improved GOP with predefined thresholds to find mispronounced segments [13], some work further incorporates a GOP-based mispronunciation detector with an EP network to boost performance [26][29], and some work utilizes the log-likelihood ratio or posterior probability vectors as input features [14][24][25] to some discriminative classifiers such as the support vector machine (SVM). Also, much work on pattern discovery has adopted posteriorgrams as the features for further processing. Some work derives the posteriorgram with GMM trained on the target corpus [20], and some with MLP trained on a separate large corpora [21].
We derive the UPP features as shown in Figure 2: we train an MLP with a large multi-speaker corpora of mixed languages [30]. The MLP has softmax output layer and the output target is the set of all phoneme units for the mixed languages, with the input being the MFCC feature vectors for all signal frames of the mixed language corpora. With this MLP we transform each frame of MFCC feature into a vector of posterior probabilities of all phonemes in the mixed languages. This is the UPP feature vector.

The motivation behind utilizing an MLP trained with multi-speaker corpora of mixed languages for posteriorgram feature extraction is to bring the information of acoustic space partitioning with phoneme units of multiple languages into the under-resourced task [21][30], which here is EP analysis. As illustrated in Figure 3, in the upper left the acoustic instances of two different pronunciation patterns A and B are scattered over the acoustic space (represented by MFCCs). Because they are produced by many speakers, the speaker variation and pronunciation variation are mixed together. This condition reflects the situation of most corpora collected in the language learning context, which consists of speech produced by learners with a wide range of native languages (L1s) and accents. The posterior probability estimator, as in the middle left of Fig. 3, or the MLP in the middle of Fig. 2 in our implementation, is then trained based on the instances from the labeled, mixed-language corpora, focusing on distinguishing different pronunciations. Therefore in the lower left of Fig. 3 patterns A and B become easier to distinguish in the posterior space (represented by UPPs), although they are produced by different speakers. However, in the testing phase (Figure 3, upper right), we do not know which instance belongs to which pattern. By borrowing the posterior probability estimator trained on the annotated, multi-speaker, mixed-language corpora from the left, we similarly map the instances from the acoustic space to the posterior space. Thus doing, we reduce speaker variation while preserving the traits of pronunciation variation, which is the key for both supervised EP detection and unsupervised EP discovery.

V. SUPERVISED DETECTION OF PRONUNCIATION ERROR PATTERNS

We consider the EPs as the pronunciation variations for each corresponding canonical pronunciation. With the AMs of the EPs as presented below, phoneme-level forced alignment can be performed with learners’ recordings. This can be done as shown in Fig. 4, in which we expand the phoneme sequence of the orthographic transcription of the utterance into a network of canonical pronunciations and EPs, each of which is an AM. We also insert a short-pause (sp) model between every two syllables to capture hesitations made by learners. Those surface pronunciations with maximum likelihood are then automatically chosen during forced alignment. Because the EPs used here include not only phoneme-level substitutions, but also insertions and deletions, these types of errors can all be detected with this scenario.

Although EPs coded “099” do not represent specific acoustic patterns, they can be considered phoneme-specific anti-models. Assume a certain phoneme possesses no EPs except
“099”: the “099” AM is then simply the collection of all the mispronunciations of this phoneme. Hence comparing the likelihoods of the canonical model and the anti-model is actually verifying the correctness of the pronunciation of a certain phoneme segment using a likelihood ratio test.

A. Acoustic modeling for EPs

The above framework requires an AM for each EP. However, the derivation of the AMs for the EPs is not as straightforward as it may seem. If we train an AM set directly using the training set of learners’ recordings with the corpus mentioned above, many EPs possess insufficient data for AM training, due to the very low percentage of mispronunciations as shown in Table I further distributed across many EPs for each canonical phoneme. In some work on EP detection, all EPs are described with the phonemes of the target language, and thus can directly utilize the AMs of the target language trained with a corpus of native speakers[23]. In our case, however, although the target language is Mandarin Chinese, some of the EPs are described using phonemes from other languages such as English. We thus adopt more elaborate approaches as reported below.

We proposed recently a special framework to explicitly derive the AM of each EP [26]. Based on the description of each EP given by the language teachers, as similar to a certain Chinese or English phoneme, we duplicated the corresponding Chinese or English phoneme models trained from some other Chinese and English corpus, both produced by native speakers, as the initial EP models. This is referred to as “heterogeneous model initialization” [26]. Then cascaded adaptation [31], which includes three stages: global Maximum Likelihood Linear Regression (MLLR), class-based MLLR and Maximum A Posteriori (MAP) adaptation cascaded in sequence, was performed with the very limited adaptation data for the EPs, as mentioned in Section III-A, after model initialization. The copies of existing Chinese or English phoneme models are therefore transformed into EP models.

In Figure 5 we further illustrate the notion of the above model initialization and adaptation procedures for deriving EP AMs, following the conceptual plot of the pronunciation space from Fig. 1. As we explained in Chapter III-B and the example in Fig. 1, the realizations of an EP may be anywhere between the canonical pronunciation, and the pronunciations used to describe their characteristics. Therefore it is not easy to determine whether it is better to adapt from Chinese /l/ (referred to as “heterogeneous initialization”) or English /l/ (heterogeneous initialization, as mentioned above) to create the model of the “l_010” EP. Nevertheless, in either case the model adaptation procedure does mitigate the difference between the EP realizations and the training data for Chinese and English AMs. Yet in a recent work [26] it has been shown that the heterogeneous initialization ensures that the models of the EPs corresponding to the same canonical phoneme are intrinsically different from each other, and thus yields better discriminability. In the area of CAPT, AM adaptation has been widely adopted for alleviating the speaker or environment mismatch between the learner’s voice and the training or reference set for fairer comparison or evaluation [32][33]. But the main purpose of model adaptation here is to create the AMs which better capture the characteristics of EPs.

After the EP models are successfully built, we use them to construct the pronunciation network as in Fig. 4 for maximal-likelihood alignment, and the surface pronunciation of learners’ recordings can thus be determined. This is the baseline system in the experiments reported below.

B. EP detection framework based on the hybrid approach

One popular approach to incorporate addition features or classifiers into the conventional HMM/GMM speech recognition framework is the hybrid approach [34]. The hybrid approach has long attracted much attention and has yielded substantial improvements in the speech recognition community. The recent success of deep neural networks in speech recognition [35] was also derived from this approach, in which the dramatically increased number of layers of MLP was shown to better exploit the opportunities offered by “Big Data”.

Based on the hybrid approach, we propose the framework for EP detection as illustrated in Fig. 6. We start by performing the first pass of Viterbi decoding on the learners’ utterances using the EP AM set as described in Sec.V-A and the pronunciation network (Fig. 4) to obtain more precise time boundaries. Then a second pass of Viterbi decoding is performed given the estimated segment boundaries, taking into account the scores from both the EP AMs (using 39-dimensional MFCCs, c0 to c12 plus derivatives and accelerations, as the frame-level feature vector) and the MLP-based EP classifiers (using MFCCs, UPPs proposed here, or different variants of UPPs as the frame-level feature vector). We assume the UPPs may be complementary to MFCCs, just as the MLPs to HMM/GMMs, and this will be verified below. Finally, we further measure the confidence of the output to decide whether to return the EP diagnosis results to the learners. This yields the final EP detection result of each phoneme segment.
In the second pass Viterbi decoding, the scores from the EP AMs and EP classifiers are integrated as follows. Let \( e_p^i \) be the \( i \)-th EP for a phones \( p, i = 0, 1, 2, \ldots, N_p \), where \( e_p^0 \) denotes the canonical pronunciation and \( N_p \) is the total number of EPs for phone \( p \). For each frame at time \( t \), the acoustic score

\[
S(t, e_p^i) \ \text{w.r.t.} \ e_p^i \text{ is}
\]

\[
S(t, e_p^i) = S_g(x_t | e_p^i) + S_d(e_p^i | y_t),
\]

where \( S_g(x_t | e_p^i) \) is the log-likelihood score of \( e_p^i \) given by the EP AMs for the frame-level feature vector of MFCC, \( x_t \), and \( S_d(e_p^i | y_t) \) is the posterior probability score of \( e_p^i \) given by the EP classifiers, for the frame-level feature vector \( y_t \). (In this paper, \( y_t \) can be MFCCs, UPPs or its variants. As will be clear below, the relative importance between \( S_g(x_t | e_p^i) \) and \( S_d(e_p^i | y_t) \) will be tuned by weighting the EP classifiers. The EP AMs are exactly the same as in the baseline approach as described in Sec.V-A; the EP classifier is described in detail below.

C. Hierarchical MLPs as the EP classifiers

Many powerful classifiers such as the SVM have been utilized in much CAPT works [14][24][25][36][37]. Here we adopt two-level hierarchical MLPs as the EP classifiers (Fig. 6) because MLP output is a well-calibrated probability estimation well-suited to integration into the HMM/GMM decoding framework.

For each phoneme \( p \), two MLPs are trained and cascaded as illustrated in Fig. 7. The first MLP is binary, and classifies each frame-level feature vector \( y_t \) as either correct or incorrect; the second is an \( N_p \)-ary MLP, and classifies each frame identified as incorrect by the first MLP into one of the \( N_p \) EPs. We denote as “binary-MLP” the first MLP, and “EP-MLP” the second MLP. In order to take more context information into consideration, we concatenated 4 preceding and 4 following frames besides the current frame at time \( t \) as the input frame-level feature vector \( y_t \) to the MLPs. The score given by the EP classifier is

\[
S_d(e_p^i | y_t) = \begin{cases} w_{1,p} \cdot \ln(P_1(\text{correct} | y_t)) & \text{if } i=0, \\ w_{1,p} \cdot \ln(P_1(\text{incorrect} | y_t)) + w_{2,p} \cdot \ln(P_2(e_p^i | y_t)) & \text{otherwise,} \end{cases}
\]

where \( P_1(\cdot) \) is from the binary-MLP, \( P_2(\cdot) \) is from the EP-MLP, and \( w_{1,p}, w_{2,p} \geq 0 \). Note that here a distinct set of MLPs is trained for each phoneme. We also allow the weights \( w_{1,p} \) and \( w_{2,p} \) to differ for different phonemes, and allow weights of 0 in case the corresponding classifiers are ill-trained. Also, since \( w_{1,p} \) and \( w_{2,p} \) can be greater than or less than 1, the relative importance between \( S_g(x_t | e_p^i) \) and \( S_d(e_p^i | y_t) \) is also incorporated implicitly in \( w_{1,p} \) and \( w_{2,p} \). We also attempted to use the monolithic MLP for each phoneme \( p \) to derive the scores of its canonical pronunciation and EPs, but that yielded much poorer performance.

D. EP diagnosis confidence estimation

After the second pass of Viterbi decoding in 6, we further estimate the confidence of the outputs of the EP-MLP in Fig. 7. For each acoustic segment \( Y \) of phoneme \( p \) with frames \( y_1, \ldots, y_T \) in the learner’s utterance, we accumulate the frame-wise entropy of the EP diagnostic results, and then take its
negative as the EP diagnosis confidence of the segment $Y$:

$$\text{confidence}(Y, p) = (-1) \sum_{i=1}^{T} \sum_{t=1}^{N_p} -P_2(e^*_t|y_t) \log N_p P_2(e^*_p|y_t).$$

(3)

If the confidence is higher than a pre-defined threshold, we return the EP diagnostic results to the learners; otherwise, we only return the binary judgment that whether the pronunciation is correct or incorrect. This is because the EP diagnostic results are intended to offer instructions regarding how the learner’s pronunciation should be fixed. If the confidence for the EP diagnostic results is not high enough, it is better to fall back to the binary classification results; otherwise the learners may receive instructions pointing to the wrong directions, which can do more harm than good.

E. Evaluation metrics

Many different evaluation metrics have been used for CAPT. For example, the correlation coefficient is commonly used when the scores given by the evaluators are continuously valued [7][38][39][40]; when discrete or class-wise prediction is given by the evaluators, however, confusion matrix-based metrics such as Cohen’s Kappa value, the false acceptance rate (FAR) false rejection rate (FRR) or precision/recall are often used [7][11][13][14][41]. Some adopt modified or weighted versions of these basic metrics [9][18]. User studies are another popular approach for evaluating the effectiveness of a CAPT system [3][10].

In the experiments reported here, we basically follow the previously-defined hierarchy of mispronunciation detection [42] shown in Fig. 8. As is clear in the figure, for correct pronunciation there can be true acceptance (TA) and false rejection (FR), and for mispronunciation there can be true rejection (TR) and false acceptance (FA). For TR instances the EPs can be given and therefore there can be correct diagnosis (CD) and diagnostic error (DE). There are thus two types of performance metrics that should be evaluated in EP detection experiments considered here. The first type is for binary classification error: for this type of metric, if a certain True Rejection instance has low confidence score on its EP diagnosis, it is neither counted in CD nor DE.

Hierarchical structure of the metrics used in EP detection. Note in the proposed framework, if a certain True Rejection instance has low confidence score on its EP diagnosis, it is neither counted in CD nor DE.

![Hierarchical structure of the metrics used in EP detection. Note in the proposed framework, if a certain True Rejection instance has low confidence score on its EP diagnosis, it is neither counted in CD nor DE.](image)

Note that with the definition of TA, TR, FA, and FR in Fig 8, we can also have the F1 score:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}},$$

(7a)

$$\text{precision} = \frac{TA}{TA + FA},$$

(7b)

$$\text{recall} = \frac{TA}{TA + FR}.$$  

(7c)

However, the F1 score might not be a fair score in the EP detection task. Because the F1 score is calculated only with TA, FA, and FR, it in fact ignores the quantity of True Rejection and fails to reflect its influence. Also, F1 score doesn’t consider the EP diagnosis and ignores CD and DE.

F. Experimental setup

In the experiments the Chinese phoneme models were trained using the ASTMIC Mandarin corpus of read speech recorded by 95 males and 95 females, each producing 200 utterances, with a total length of 24.6 hours; the English
phoneme models were trained on the TIMIT corpus training set [43] recorded by 462 speakers from eight dialect regions of the USA, with a total length of 3.9 hours. We chose the monophone as the phoneme model unit for both Chinese and English. Most Zhuyins of Mandarin Chinese are based on monophones except some diphthongs, for which we modified the lexicon so that a diphthong can be mapped to two or more consecutive monophone models. The AMs are 3-state HMMs, and each state is modeled by a 32-mixture GMM.

For the UPP derivation, the training corpora included both the ASTMIC Mandarin corpus and the TIMIT English corpus training set as mentioned above. The MLP training target was the union of the monophone sets of Mandarin Chinese and English, consisting of 35 and 38 monophones respectively with a total of 73, without short pause or silence.

For the hierarchical MLPs in the EP classifiers in Fig. 6 and 7, both the binary-MLP and the EP-MLP as well as the corresponding weights \( w_{1,p} \) and \( w_{2,p} \) in Eq. 2 are to be determined for each phoneme \( p \). In our experiments, the training sets of binary- and EP-MLPs were the same, i.e. the adaptation set in Table I. For MLP training, the input frame-level feature vectors are one of the following, in each case concatenated with 4 preceding and 4 following frames to take more context information into consideration:

1. MFCC (39 parameters, c0 to c12 plus first and second derivatives);
2. UPP (73 posteriors for 73 Mandarin/English monophones);
3. Logarithm of UPP (log-UPP);
4. Principal component analysis (PCA) transformed log-UPP (PCA-log-UPP). For PCA we retained 95% of the total variance; and the classification targets were as described in Sec.V-B.

One problem arose when training the binary-MLPs: there were far more correctly-pronounced than mispronounced instances. To take into account this data imbalance, we down-sampled the correctly-pronounced instances to the same size as the mispronounced instances when training the binary-MLP of each phoneme.

The numbers of hidden nodes in the MLPs were tuned on the development set to minimize the frame-level FAR+FRR for binary-MLPs, and to minimize the frame-level DER for EP-MLPs. After the optimal MLPs of each phoneme were determined, we embedded the MLPs into the proposed structure in Fig. 7 and proposed framework in Fig. 6, and the weights \( w_{1,p} \) and \( w_{2,p} \) in Eq. 2 were then tuned to minimize the segment-level AER of the overall system on the development set. The weights \( w_{1,p} \) and \( w_{2,p} \) were allowed to be 0 in case of ill-trained MLPs due to limited training data. Finally, the thresholds for the confidence verification in Section V-D were also tuned for each phoneme \( p \) on the development set.

G. Experimental results and discussion

Table IV contains the experimental results on the test set in Table I, including the performance achieved using the proposed approach (rows (2)–(5)) compared to the baseline (row (1)) and to the human expert (row (6)). Note for the DER, since the denominators (TR) were different for different experimental settings, we also provide the raw numbers of TR (except for the result of human expert, since it is not calculated on the test set) because the numbers of DER may not reflect the whole picture of the EP detection performance.

First, note that in all cases including automatic detector and human annotator, the FARs are always much higher than FRRs; that is, for either machine or human, the testing segments are prone to be judged correct even if they actually belong to some EPs. An explanation for this phenomena is that, in addition to the fact that all EPs are close to the canonical phonemes and thus hard to distinguish, the learners producing our corpus had already undergone basic training in Mandarin Chinese. When the percentage of mispronunciation is low, the automatic detectors may under-estimate the likelihood of mispronounced instances; and when the proficiency level of the learners are actually not too bad, it may also influence the impression of human experts, who thus tend to falsely accept their mispronunciations. Fortunately, as well known in CAPT, higher FARs may be less harmful than higher FRRs since false rejections could discourage learners.

Rows (2)–(5) are the results of the proposed approach by combining scores from EP AMs and EP classifiers (i.e. hierarchical MLPs in our implementation) with different feature vectors as the input of EP classifiers. First, all the proposed approaches (rows (2)–(5)), regardless of the feature used, achieved significantly lower DER than the baseline (row (1)), especially when using UPP as the input features (row (3)), and also lower FAR. Note the numbers of TR of the baseline and proposed approaches (rows (1)–(5)) are very close, and therefore the numbers of DER actually provided clear comparison of performance. Despite the FRR was only slightly improved, the overall performance (AER) was much better than the baseline (row (1)) in every proposed approach for all features tested, and using the proposed UPP (row (3)) we achieved the best results in the supervised EP detection experiments. However, it is also clear that there is still a wide gap between the proposed approaches (rows (2)–(5)) and the human expert (row (6)), indicating the space for possible improvement in the future.

We further examined the effectiveness of confidence verification step. First, if without this step, the improvement of EP diagnosis results (DER) are in general not significant w.r.t. baseline. On the other hand, after applying confidence verification step, we filtered out 48.2%, 42.1%, 51.3% and 42.1% TR instances with low confidence and input features being MFCC, UPP, log-UPP and PCA-log-UPP respectively. The DERs as shown in Table IV are thus significantly reduced.

<table>
<thead>
<tr>
<th>Features of EP classifiers</th>
<th>FAR</th>
<th>FRR</th>
<th>DER (TR)</th>
<th>AER</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) None (baseline)</td>
<td>47.1%</td>
<td>12.9%</td>
<td>31.0% (1032)</td>
<td>30.2%</td>
</tr>
<tr>
<td>(2) MFCC</td>
<td>48.3%</td>
<td>12.5%</td>
<td>21.3% (1087)</td>
<td>20.4%</td>
</tr>
<tr>
<td>(3) UPP</td>
<td>46.2%</td>
<td>12.4%</td>
<td>19.8% (1068)</td>
<td>26.1%</td>
</tr>
<tr>
<td>(4) log-UPP</td>
<td>46.2%</td>
<td>12.3%</td>
<td>24.8% (1660)</td>
<td>27.8%</td>
</tr>
<tr>
<td>(5) PCA-log-UPP</td>
<td>45.9%</td>
<td>12.5%</td>
<td>20.6% (1668)</td>
<td>28.3%</td>
</tr>
<tr>
<td>(6) Human expert</td>
<td>29.4%</td>
<td>4.2%</td>
<td>14.3% (N/A)</td>
<td>11.7%</td>
</tr>
</tbody>
</table>
H. Complementarity analysis for the EP classifiers and EP AMs in the proposed framework

To understand how well the EP classifiers were trained and selected, we further counted the number \( C_p \) of phonemes whose EP classifiers were actually activated, i.e. \( w_{1,p} > 0, w_{2,p} > 0 \):

\[
C_p = \text{Count}\{ p \mid w_{1,p} > 0, w_{2,p} > 0 \}; \quad (8)
\]

and evaluated the average number of EPs, \( \bar{N}_p^* \), of those phonemes with activated EP classifiers:

\[
\bar{N}_p^* = \frac{1}{C_p} \sum_{p} N_p^* \quad (9)
\]

These statistics are listed in Table V for different sets of features used in the EP classifiers. First, we found that out of the 39 sets of hierarchical MLPs, only 12 were activated, no matter which set of input features is used. For the other 27 phonemes the EP classifiers were ill-trained and not used at all, or simply the baseline EP AMs were used. Next, for those phonemes with activated EP classifiers, there were much more number of EPs (5.4–5.8) than average (3.9) as reported in Sec. III-B. This implies despite there were 27 EP classifiers that were trained too badly to be activated by the system, those 12 activated EP classifiers were actually selected to tackle some of the most confusing groups of EPs. This is why in Table IV the 12 activated EP classifiers did offer great improvements to the overall performance, especially EP diagnosis (DER). This verified that the EP classifiers proposed here are complementary to the baseline of EP AMs.

In Table VI we further illustrate the detailed results, including FAR, FRR, DER, AER and the relative reductions in error rates (in parentheses) compared to the baseline, for the 12 phonemes with activated EP classifiers using UPP as input features. First, we can see that the phonemes /b/, /a/, /o/ and /i/ have relatively high FAR and low FRR. One possible explanation is that they are very common pronunciations existing in many languages, and their mispronunciations are thus either very rare or very subtle to distinguish. In such cases our automatic detector simply accept most of their instances as correct pronunciation. What is more, note that the 0.0% DER of phonemes /b/, /a/ and /o/ were in fact not because the system did not make any erroneous diagnosis, but because all their EP diagnosis could not pass the confidence verification we mentioned in Section V-D, and thus no diagnostic feedbacks were given at all. This shows how the confidence verification step can prevent learners from receiving incorrect diagnostic instructions as well as how this is reflected in our evaluation metrics.

When we exclude the above 4 phonemes (/b/, /a/, /o/ and /i/) and focus on the relative improvement, we can see that the phonemes /zh/, /s/ and /iu/ corresponding to 8, 5 and 5 EPs respectively in Table VI, have the largest relative reduction in DER as well as AER. This illustrates how much the extra discriminating power the EP classifiers (hierarchical MLPs) has introduced into the automatic detector, especially for those phoneme units with more EPs than average.

<table>
<thead>
<tr>
<th>Phoneme</th>
<th>FAR (%)</th>
<th>FRR (%)</th>
<th>DER (%)</th>
<th>AER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>7.5 (6.5)</td>
<td>0.0 (100.0)</td>
<td>0.0 (14.6)</td>
<td>12.9 (5.9)</td>
</tr>
<tr>
<td>zh</td>
<td>28.8 (42.9)</td>
<td>35.5 (21.4)</td>
<td>16.7 (59.7)</td>
<td>27.0 (24.1)</td>
</tr>
<tr>
<td>ch</td>
<td>30.4 (4.4)</td>
<td>12.9 (18.5)</td>
<td>33.3 (27.4)</td>
<td>32.2 (11.8)</td>
</tr>
<tr>
<td>sh</td>
<td>32.5 (17.1)</td>
<td>24.8 (2.4)</td>
<td>47.6 (5.9)</td>
<td>33.7 (5.9)</td>
</tr>
<tr>
<td>r</td>
<td>38.7 (2.6)</td>
<td>14.3 (18.5)</td>
<td>30.8 (2.4)</td>
<td>34.6 (5.9)</td>
</tr>
<tr>
<td>s</td>
<td>36.4 (20.0)</td>
<td>9.24 (3.6)</td>
<td>0.0 (0.0)</td>
<td>15.2 (48.0)</td>
</tr>
<tr>
<td>a</td>
<td>72.0 (0.0)</td>
<td>4.3 (18.4)</td>
<td>0.0 (100.0)</td>
<td>25.4 (48.7)</td>
</tr>
<tr>
<td>o</td>
<td>72.7 (19.7)</td>
<td>9.3 (7.6)</td>
<td>0.0 (100.0)</td>
<td>25.4 (25.2)</td>
</tr>
<tr>
<td>en</td>
<td>38.8 (22.3)</td>
<td>26.5 (7.9)</td>
<td>35.4 (15.2)</td>
<td>33.6 (13.4)</td>
</tr>
<tr>
<td>i</td>
<td>74.1 (3.3)</td>
<td>9.1 (2.3)</td>
<td>14.3 (4.6)</td>
<td>32.7 (4.6)</td>
</tr>
<tr>
<td>u</td>
<td>67.0 (1.6)</td>
<td>12.0 (3.8)</td>
<td>22.2 (39.7)</td>
<td>33.7 (13.8)</td>
</tr>
<tr>
<td>iu</td>
<td>29.7 (2.4)</td>
<td>47.9 (0.4)</td>
<td>26.0 (57.1)</td>
<td>35.4 (26.9)</td>
</tr>
</tbody>
</table>

VI. UNSUPERVISED DISCOVERY OF PRONUNCIATION ERROR PATTERNS

As mentioned previously, EPs are naturally different for different pairs of target and native languages. Deriving EPs for each language pair requires extensive expert knowledge, huge quantities of labeled data, or reliable ASR results. Hence the need for unsupervised approaches for automatic discovery of EPs, from a corpus of learners’ voice data for given target/native languages and given conditions. This section presents a framework as the initial approach towards such a goal.

The task here is to unsupervisedly discover the EPs for each phoneme given a corpus of learner voice data. We assume the canonical transcriptions of the utterances in the given corpus are available, so forced alignments can be generated and the learner voice data is divided into segments corresponding to phonemes. We can thus focus on one phoneme at a time: each time we are given a set of acoustic segments corresponding to a specific phoneme, and the goal is to divide this set into several clusters, each of which corresponds to an EP. Furthermore, because the percentage of correct pronunciations in our corpus far exceeds that of the mispronunciations, we here take into account the data imbalance by excluding correctly pronounced segments before clustering.
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2014.2387413, IEEE/ACM Transactions on Audio, Speech, and Language Processing

With the above settings, our task is as illustrated in Fig. 9, using the conceptual pronunciation space as in Fig. 1. Given the mispronounced instances of a certain phoneme $p$, we wish to automatically derive several clusters in an unsupervised way, each may represent a certain EP.

A. Framework overview

Figure 10 shows the proposed framework for unsupervised EP discovery from a given corpus. First we extract frame-level feature vectors $z_1, z_2, ..., z_t, ...$ for speech segments corresponding to each phoneme $p$. As will be clear later, the UPPs as described in Section IV can be good candidates of the frame-level feature vectors. As explained in Sec. IV, with UPPs we transform the MFCC feature vectors in the more speaker-dependent acoustic space (in terms of MFCCs) into the less speaker-dependent posterior space (in terms of UPPs).

We first apply hierarchical agglomerative clustering (HAC) to merge adjacent frame-level feature vectors $z_1, z_2, ..., z_t, ...$ within a signal segment corresponding to each specific phoneme $p$ into a tree-structured hierarchy, based on how this segment is divided into $M_p$ sub-segments [22]. Each sub-segment is composed of acoustically similar frames. This is shown in the upper half of Fig. 10. The number of sub-segments $M_p$ is the same for all segments corresponding to a specific phoneme $p$, but can be different for different phonemes. We take the average of the frame-level feature vectors in each of these $M_p$ sub-segments, yielding $M_p$ mean vectors $Z_1, Z_2, ..., Z_{M_p}$ respectively. Then we concatenate these averaged feature vectors into super-vector $o = [Z_1^T, Z_2^T, ..., Z_{M_p}^T]^T$ as the segment-level feature vector, as shown in the central part of Fig. 10. The segment-level feature vector $o$ for all speech segments corresponding to phoneme $p$ are then clustered into different EPs by an unsupervised algorithm. In this preliminary work we utilized K-means and GMM with the minimum description length (MDL) principle [44] for unsupervised clustering, as shown in the lower part of Fig. 10.

B. Hierarchical Agglomerative Clustering (HAC) and Segment-level Feature Vectors

Because the instances of EPs are signal segments, we need to define segment-level features for them; and since we perform EP discovery separately for each phoneme $p$, the length of these feature vectors for a specific phoneme $p$ must be identical. However, the number of frames in different segments for a given phoneme varies. We cannot simply concatenate all the frame-level features $z_t$ into a segment-level feature vector. Nor can we average all the frame-level feature vectors in a segment for a phoneme into one vector, because the difference between EPs and their corresponding canonical pronunciation can be very subtle, often differing only in sub-segmental realization, and averaging all frames within a speech segment may destroy such subtle evidence. The HAC algorithm used here automatically arranges the frames in a speech segment into a tree-structured hierarchy, in which similar adjacent frames are clustered together in lower layers, while relatively dissimilar adjacent clusters are merged in higher layers. In this way different similarity thresholds among frames result in different numbers of sub-segments.

The HAC algorithm we adopted is as follows [45]. Let $B = (t_0, ..., t_{M_p})$ be the set of boundaries dividing a given segment

---

Fig. 9. The mispronounced instances of a certain phoneme $p$ in the conceptual pronunciation space as in Fig. 1, and the possible EPs automatically discovered as considered here.

Fig. 10. Proposed framework for unsupervised EP discovery.
of \( L \) frames into \( M_p \) sub-segments with \( 0 = t_0 < t_1 < \ldots < t_{M_p} = L \). The \( m \)-th sub-segment \( (z_{t_{m-1}}+1, \ldots, z_{t_m}) \) ends at \( t_m \). The sum of squared error (SSE) when representing all frames in a sub-segment by their means for the boundary set \( B \) is

\[
SSE(B) = \sum_{m=1}^{M_p} \sum_{t=t_{m-1}+1}^{t_m} |z_t - \bar{Z}_m|^2, \tag{10}
\]

where \( \bar{Z}_m \) is the mean vector of the \( m \)-th sub-segment. The HAC algorithm is used to find the boundary set \( B \) that yields the smallest SSE for any chosen number of sub-segments. The initial boundary set \( B_0 \) is simply \( \{0, 1, 2, \ldots, L\} \), i.e. every frame is a distinct sub-segment. At each iteration \( j \), the algorithm chooses to merge two neighboring sub-segments (by deleting one internal boundary from \( B_{j-1} \)) that yields the minimum increment of SSE, \( \Delta SSE_j = SSE(B_j) - SSE(B_{j-1}) \), where \( B_j \) is the resulting boundary set after merging. The algorithm stops when there is only one sub-segment left, which includes all the \( L \) frames [45].

After the above algorithm stops, we have a tree-structured hierarchy based on the merging order of sub-segments. By choosing a threshold of SSE, \( \lambda \) (this is shown on the left of the hierarchy in the upper left of Fig. 10), we can hence obtain the boundary set \( B \) that partitions the original segment into any desired number \( M_p \) of sub-segments. As a result, by tuning \( \lambda \) and \( M_p \) we can optimize the granularity of the segment-level feature vectors for each phoneme \( p \). In this way the HAC can not only unify the dimensionality of the segment-level feature vectors for all the segments corresponding to each specific phoneme \( p \), but also offers the opportunity to properly choose the number of sub-segments \( M_p \), such that the differences among EPs are better retained.

C. Unsupervised Clustering Algorithms for EP Discovery

After all signal segments for a specific phoneme \( p \) are represented as segment-level feature vectors \( o \) based on the mean vector \( \bar{Z}_i, i = 1, \ldots, M_p \), as shown in Fig. 10, we can then cluster these segment-level feature vectors into EPs. In this preliminary work we used two algorithms for EP clustering: K-means and GMM with the minimum description length principle (GMM-MDL) [44]. For K-means we need to know the number of clusters. For this reason we assume the number of clusters \( k \) is known, which is the number of EPs of each phoneme summarized by the language teachers. This is not realistic for an unsupervised EP discovery task though. Several different distance measures are considered in the K-means algorithm here: Euclidean distance \( d_{euc}(o,o') \), cosine distance \( d_{cos}(o,o') \), and symmetric KL divergence \( d_{kl}(o,o') \), where

\[
d_{cos}(o,o') = 1 - \frac{o \cdot o'}{|o||o'|}, \tag{11}
\]

\[
d_{kl}(o,o') = \frac{1}{2} \sum_{i=1}^{D} (o_i \cdot \log \frac{o_i}{o_i'}) + o_i' \cdot \log \frac{o_i'}{o_i}, \tag{12}
\]

where \( o \) and \( o' \) are the segment-level feature vectors with dimensionality \( D \), and \( o_i, o_i' \) are their \( i \)-th component respectively. The symmetric KL divergence was tested with UPP only in our experiments, since UPP is a properly normalized probability distribution.

In a more ideal scenario, the number of clusters (or EPs) should be learned from data. For this purpose we use the GMM-MDL algorithm. We trained one GMM for each phoneme \( p \), and then perform maximum-likelihood (ML) classification to assign instances to clusters. The GMM-MDL algorithm is capable of estimating the optimal number of Gaussians in GMM, which is the number of EPs here. The optimal number of Gaussians is obtained by maximizing the following objective function:

\[
F(O_p, \theta_p) = \log P_r(O_p|\theta_p) - \frac{1}{2} |\theta_p| \log(|O_p| D_p), \tag{13}
\]

where \( \theta_p \) is the parameter set of the GMM for phoneme \( p \), \( O_p \) the set of segment-level feature vectors for phoneme \( p \) of dimensionality \( D_p \), with size \( |O_p|, |\theta_p| \) the total number of continuously-valued free variables to specify \( \theta_p \):

\[
|\theta_p| = G_p(1 + D_p + \frac{(D_p+1)D_p}{2}) - 1, \tag{14}
\]

where \( G_p \) is the number of Gaussians. Eq. (14) comes from the fact that for each Gaussian there are 1 prior probability, \( D_p \) values for the mean, and \( \frac{(D_p+1)D_p}{2} \) variables for the covariance matrix. Because the \( G_p \) priors sum to one, the overall degree of freedom is reduced by 1. In Eq. (13) the first term on the right hand side is the log-likelihood, and the second term represents the model complexity. Since the number of Gaussians (number of EPs here) is also a free parameter in Eq. 13, it is hence optimized altogether based on the balance between the two considerations.

D. Evaluation Metrics

There are many different metrics for evaluating clustering algorithms. Cluster purity is a good example, although it tends to favor a larger number of clusters. Here we adopt the Rand Index [46] for its balance between the similarity within clusters and dissimilarity among different clusters.

We first define the pair-wise true acceptance (TA’), true rejection (TR’), false acceptance(FA’), and false rejection (FR’) for clustering tasks based on all instance pairs as in Table VII [21]. For example, if an instance pair belongs to the same cluster in both the reference and the prediction result, it is counted as one in TA’, and so on. We see that TA’ and TR’ represent respectively the within-cluster and between-cluster accuracies. The Rand Index is then defined as

\[
RI = \frac{TA' + TR'}{TA' + TR' + FA' + FR'}. \tag{15}
\]

Since the mispronounced segments for each phoneme \( p \) are individually clustered for EPs, we report the Average Rand Index (ARI) over all phonemes \( p \) for the phoneme set \( P \):

\[
ARI = \frac{1}{|P|} \sum_{p \in P} RI(p), \tag{16}
\]

where \( RI(p) \) is the Rand Index for phoneme \( p \) and \( |P| \) the total number of phonemes considered.
The definition of true acceptance (TA'), true rejection (TR'), false acceptance (FA'), and false rejection (FR') for the evaluation of clustering tasks.

<table>
<thead>
<tr>
<th>For all instance pairs belonging to</th>
<th>Reference clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted same</td>
<td>TA'</td>
</tr>
<tr>
<td>Predicted different</td>
<td>FR'</td>
</tr>
</tbody>
</table>

E. Experimental setup

In the preliminary experiments, for the frame-level feature vectors $z_t$ in Fig. 10, besides UPP as described in Section IV, we also tested other frame-level features including logarithm of UPP (log-UPP), principal component analysis (PCA) transformed log-UPP (PCA-log-UPP), and MFCC (39 parameters, c0 to c12 plus first and second derivatives).

Three different choices of number of sub-segments $M_p$ in HAC were considered: $M_p = 1, M_p = 1$ and $M_p = M_{\text{max}}$. For $M_p = 1$ we did not divide a segment into smaller sub-segments, but simply average all frame-level feature vectors in each segment into one segment-level feature vector. For $M_p = M_{\text{opt}}$ we tuned $M_p$ for each phoneme $p$ by optimizing the performance. For $M_p = M_{\text{max}}$ we set $M_p$ to be the number of frames in the shortest segment; thus for the shortest segment we treat each frame as a sub-segment.

Besides the above three HAC settings, we also tested a common heuristic setting in which each monophone was partitioned into 3 equal length sub-segments, and each diphthong into 6. Note however that in our experiment, $M_{\text{max}}$ is actually 3 for monophone and 6 for diphthong. This means in the heuristic setting we actually have $M_p = M_{\text{max}}$, except the length of each sub-segment was manually set as equal.

F. Experimental results (I) – K-means with assumed known number of EPs

Table VIII reports the ARI using K-means, for which we assume the number of EPs for each phoneme is known, which is the number of EPs summarized by the language teachers. Different rows represent different features and different distance measures used in the K-means algorithm, except the last row for human expert, and different columns represent different sub-segmentation approaches, including HAC with different numbers of sub-segments $M_p$, and the heuristic approach (equal length segmentation). We can see that certainly there is a gap between those achieved by machine and the high consistency achieved by the human annotator.

Because there is no any prior work reported for this task [27], there is no baseline which should be compared with here. But in order to see if the difference between these numbers in the table are really significant, we performed significance testing on these results, with significance level being 5%, tentatively taking the first row in the table (MFCC with Euclidean distance) as the baseline. This significance analysis showed that only the results of using log-UPP with cosine distance and $M_p = 1$ or $M_p = M_{\text{opt}}$ was significantly better. This verified that the proposed UPPs, with proper configuration, can distinguish patterns of pronunciation and in turns discover the EPs in the posterior space, and significantly outperformed the conventional features such as MFCCs.

By comparing different columns, the results of HAC segmentation with $M_p = M_{\text{opt}}$ yielded significantly better results than $M_p = 1, M_p = M_{\text{max}}$ and the heuristic approach with all different features and distances (with the same 5% significance level). Note, though, that the results given by $M_p = M_{\text{opt}}$ is simply an upper bound of the ARI achievable by properly setting the number of sub-segments with the HAC if allowed, since one should not be able to tune the parameters for optimal performance under the unsupervised condition. Yet the fact that $M_{\text{opt}}$ gave the highest ARI in each row shows that the difference among EPs may lie in sub-segmental realizations which can be better explored by varying $M_p$ in HAC. Simply setting $M_p = 1$ yielded features that were too coarse, and partitioning each segment into $M_{\text{max}}$ sub-segments may lead to over-analysis and introduce noise.

Also note that the results given by HAC with $M_p = M_{\text{max}}$ are not always better than the heuristic approach. As we explained, these two approaches were based on the same number of sub-segments, and only differed in that the sub-segments of the former had variable lengths while the latter fixed. This result thus indicates that the joint optimization of the number of sub-segments as well as their lengths is necessary for the best possible performance.

G. Experimental results (II) – GMM-MDL with automatically estimated number of EPs

Table IX shows the results of ARI using GMM-MDL with an automatically estimated number of EPs for each phoneme. The numbers in parentheses are the automatically estimated number of EPs for each phoneme. The results which are significantly better than the first row (using MFCC and Euclidean distance) with 5% significance level are marked in bold.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Distance measure</th>
<th>HAC (M_p = 1)</th>
<th>HAC (M_p = M_{\text{opt}})</th>
<th>HAC (M_p = M_{\text{max}})</th>
<th>Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>$d_{\text{cos}}$</td>
<td>57.5</td>
<td>58.5</td>
<td>57.1</td>
<td>57.6</td>
</tr>
<tr>
<td>MFCC</td>
<td>$d_{\text{euc}}$</td>
<td>57.2</td>
<td>58.5</td>
<td>57.6</td>
<td>57.5</td>
</tr>
<tr>
<td>UPP</td>
<td>$d_{\text{cos}}$</td>
<td>56.5</td>
<td>57.3</td>
<td>56.4</td>
<td>56.5</td>
</tr>
<tr>
<td>UPP</td>
<td>$d_{\text{euc}}$</td>
<td>58.5</td>
<td>57.9</td>
<td>58.2</td>
<td>57.8</td>
</tr>
<tr>
<td>log-UPP</td>
<td>$d_{\text{cos}}$</td>
<td>58.5</td>
<td>58.1</td>
<td>58.2</td>
<td>57.8</td>
</tr>
<tr>
<td>log-UPP</td>
<td>$d_{\text{euc}}$</td>
<td>58.4</td>
<td>59.2</td>
<td>58.5</td>
<td>58.5</td>
</tr>
<tr>
<td>PCA-log-UPP</td>
<td>$d_{\text{cos}}$</td>
<td>58.5</td>
<td>58.4</td>
<td>58.2</td>
<td>57.4</td>
</tr>
<tr>
<td>Human annotator</td>
<td>58.0</td>
<td>58.1</td>
<td>58.1</td>
<td>57.3</td>
<td>57.3</td>
</tr>
</tbody>
</table>

Table IX shows the results of ARI using GMM-MDL with an automatically estimated number of EPs for each phoneme. The numbers in parentheses are the automatically estimated number of EPs minus that summarized by human experts, averaged over all phonemes. The ARIs in Table IX were lower than those of Table VIII, obviously due to the lack of expert knowledge about the number of EPs.

We have also performed significance tests on these results, with significance level being 5%. Similarly, there is no baseline with which the results here should be compared to. If we take the first row using MFCC as the baseline, only the results of using log-UPP with HAC setting $M_p = M_{\text{max}}$ and
heuristic sub-segmentation approach were significantly better, highlighting the superiority of UPPs as compared to MFCCs.

On the other hand, by comparing different columns, using HAC segmentation with $M_p = M_{opt}$ yielded significantly better results than $M_p = 1$, $M_p = M_{max}$ and the heuristic approach in almost every case, except that between $M_p = M_{opt}$ and $M_p = 1$ the heuristic approach with log-UPP. And same as the previous experiment, the results given by HAC with $M_p = M_{max}$ were not always better than the heuristic approach. This again highlights the importance of dividing the signal segments into proper number of sub-segments with flexible lengths using HAC.

Note that the UPP and log-UPP features yielded 1 to 3 more automatically derived EPs than human-defined EPs in average, while PCA-log-UPP also yielded slightly more EPs. In other words, with UPP or its variants the machine is able to perform slightly finer clustering, while there are some patterns with only subtle differences that human experts may consider the same. In contrast, MFCC resulted in a lower number of clusters; this further shows the superior discriminating power of UPP in discovering EPs.

### TABLE IX

<table>
<thead>
<tr>
<th>Feature</th>
<th>Sub-segmentation approach</th>
<th>HAC ($M_p$; number of sub-segments)</th>
<th>Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$M_p = 1$</td>
<td>$M_p = M_{opt}$</td>
</tr>
<tr>
<td>MFCC</td>
<td>53.3</td>
<td>56.3</td>
<td>52.2</td>
</tr>
<tr>
<td></td>
<td>(-0.6)</td>
<td>(-0.6)</td>
<td>(-1.2)</td>
</tr>
<tr>
<td>UPP</td>
<td>24.5</td>
<td>36.0</td>
<td>34.8</td>
</tr>
<tr>
<td></td>
<td>(3.0)</td>
<td>(2.5)</td>
<td>(2.1)</td>
</tr>
<tr>
<td>log-UPP</td>
<td>34.9</td>
<td>36.7</td>
<td>54.6</td>
</tr>
<tr>
<td></td>
<td>(2.1)</td>
<td>(1.7)</td>
<td>(0.8)</td>
</tr>
<tr>
<td>PCA-log-UPP</td>
<td>54.7</td>
<td>55.4</td>
<td>53.1</td>
</tr>
<tr>
<td></td>
<td>(0.3)</td>
<td>(0.2)</td>
<td>(-0.2)</td>
</tr>
</tbody>
</table>

### H. Analysis for an example set of automatically discovered EPs

Below we try to analyze a typical set of examples of automatically discovered EPs in the log-UPP space, with which significantly better results than MFCC were obtained. Fig. 11 illustrates the statistics of the three automatically discovered EPs for the mispronounced instances of the phoneme Chinese /b/ using GMM-MDL with log-UPP. The three expert-defined EPs (b_010, b_020 and b_099) for Chinese /b/ by three different colors, in each of the automatically discovered EPs. Visualized in the lower part are respectively the displacement in each dimension of the log-UPP space between the centroids of the instances of three automatically discovered EPs and that of the correctly pronounced instances of Chinese /b/, as conceptually illustrated in Fig. 12.

**Fig. 11.** The statistical analysis of the three automatically discovered EP (#1, #2 and #3) for the mispronunciation of Chinese /b/ using GMM-MDL with log-UPP. The pie charts in the upper part show the percentage of the three expert-defined EPs (b_010, b_020 and b_099) for Chinese /b/ by three different colors, in each of the automatically discovered EPs. Visualized in the lower part are respectively the displacement in each dimension of the log-UPP space between the centroids of the instances of three automatically discovered EPs and that of the correctly pronounced instances of Chinese /b/, as conceptually illustrated in Fig. 12.

The displacement in each dimension of log-UPP, which is for each Chinese or English phoneme, is evaluated between the centroid of instances of the automatically discovered EPs and that of the correctly pronounced instances of Chinese /b/, as illustrated in Fig. 12. Then we visualize the displacements in all dimensions of log-UPP in the lower part of Fig. 11. Here each bar represents a certain dimension in log-UPP, or a specific Chinese or English phoneme. The darker the bar is, the higher the value is.

Note here the displacements are evaluated in each dimension of the log-UPP space, while each dimension of log-UPP represents the log-posterior probability of the input frame with respect to a certain Chinese or English phoneme. As a result, the displacement in each dimension of the log-UPP between two centroids is in fact the log-posterior probability ratio between the centroids of instances of the mispronunciation and correct pronunciation with respect to the corresponding discovered EP #2 in the middle, the expert-defined EP “b_010” was less dominating and the percentages of the other two expert-defined EPs were higher. On the other hand, the automatically discovered EP #3 on the right is definitely related to the expert-defined EP “b_020”, that is, the learners were judged to have mispronounced Chinese /b/ as voiced.

Next, we calculate the displacement in each dimension of log-UPP, which is for each Chinese or English phoneme, between the centroid of instances of the automatically discovered EPs and that of the correctly pronounced instances of Chinese /b/, as illustrated in Fig. 12. Then we visualize the displacements in all dimensions of log-UPP in the lower part of Fig. 11. Here each bar represents a certain dimension in log-UPP, or a specific Chinese or English phoneme. The darker the bar is, the higher the value is.
Chinese or English phoneme \( p \):

\[
\log(\frac{Pr(p|"\text{error}"))}{Pr(p|"\text{correct}"))} = \log[\frac{Pr(p|\text{error}")}{Pr(p|\text{correct}")}] \\
(17)
\]

One may wonder that this equation looks similar to GOP. But note that the direction of comparison is reversed: in the calculation of GOP we compare the learner’s pronunciation to the canonical pronunciation, while the displacement of log-UPP centroid is to compare each phoneme in the phone set to learner’s incorrect and correct pronunciation. The higher the ratio is (or the darker the bar is), the more this EP has its UPP centroid is to compare each phoneme in the phone set to the canonical pronunciation, while the displacement of log-posterior probability ratios with respect to Chinese or English phoneme \( p \).

We interpret the results in the lower part of Fig. 11 as follows:

1) For the automatically discovered EP #1, we see that its log-posterior probability ratios with respect to Chinese plosives and English stops are relatively high. This means this EP is composed of segments that sound even more like a plosive or a stop than the canonical pronunciation of Chinese /b/. This coincides with the fact that this EP is composed of mispronounced instances of /b/ that sound similar to /p/, as annotated by the human experts (b_010).

2) For the automatically discovered EP #2, it is hard to induce any obvious trend out of the log-posterior ratios. This is parallel to the fact that this EP is composed of a mix of different EPs defined by the human experts.

3) For the automatically discovered EP #3, we see that its log-posterior probability ratios with respect to Chinese and English vowels are higher. This again coincides with the fact that this EP is dominated by instances similar to voiced /b/’s as the human expert annotations suggested (b_020).

The above analysis shows that the proposed UPP features actually offer further insights for the mispronunciations produced by language learners, in addition to being able to enhance the performance of the unsupervised EP discovery task. We look forward to seeing more applications that incorporate UPP in CAPT systems.

**VII. CONCLUSIONS**

In this paper we consider both supervised detection and unsupervised discovery of pronunciation EPs in computer-aided language learning. We propose new frameworks for both supervised detection and unsupervised discovery of pronunciation EPs with empirical analysis over different approaches.

In supervised EP detection we integrate the scores from both HMM-based EP models and MLP-based EP classifiers with a two-pass Viterbi decoding architecture. We use EP classifiers in Viterbi decoding to encompass different aspects of EP detection, while maintaining flexibility for fine tuning. Experimental results showed that this integration effectively improved EP detection, and achieved significantly lower EP diagnostic error rates. In unsupervised EP discovery, we use the hierarchical agglomerative clustering (HAC) algorithm to divide speech segments corresponding to a phoneme into subsegments, to construct segment-level feature vectors retaining the sub-segmental behavior of the signal within the segments. Preliminary experimental results showed the proposed framework to be a good initial approach toward unsupervised discovery of EPs, although there is still a long way to go to approach the performance of human experts.

In both tasks we propose to utilized the universal phoneme posteriorgram (UPP), derived from a multi-layer perceptron (MLP) trained on corpora of mixed languages, as a set of very useful features to reduce speaker variation while maintaining pronunciation variation across speech frames. Experimental results showed that the UPP not only yielded the lowest error rates in both supervised detection and unsupervised discovery tasks, but offered new insight for mispronunciation analysis in computer-aided language learning.

**REFERENCES**


—, “Improved approaches of modeling and detecting error patterns with empirical analysis for computer-aided pronunciation training,” in Proc. ICASSP 2012.

—, “Toward unsupervised discovery of pronunciation error patterns using universal phoneme posteriorgram for computer-assisted language learning,” in Proc ICASSP 2013.


C. Yeh, C. Huang, and L. Lee, “Bilingual acoustic model adaptation by unit merging on different levels and cross-level integration,” in Proc. INTERSPEECH 2011.

C. Huang, F. Zhang, and F. Soong, “Improving automatic evaluation of mandarin pronunciation with speaker adaptive training (sat) and mlr speaker adaption,” in Proc. ISCSLP’08, pp. 1–4.


Yow-Bang Wang was born in 1984. He received his B.S. degree from Dept. of Electrical Engineering, National Taiwan University in 2006, M.S. degree from Graduate Institute of Communication Engineering, National Taiwan University in 2008, and Ph.D. degree from Graduate Institute of Electrical Engineering, National Taiwan University in 2014. He is currently a Senior Research Scientist in Nuance Communications, Inc.. His research interests include acoustic modeling, computer-assisted language learning and Mandarin Chinese tone recognition.

Lin-shan Lee, (F93) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

He has been a Professor of electrical engineering and computer science at the National Taiwan University, Taipei, Taiwan, since 1982 and holds a joint appointment as a Research Fellow of Academia Sinica, Taipei. His research interests include digital communications and spoken language processing. He developed several of the earliest versions of Chinese spoken language processing systems in the world including text-to-speech systems, natural language analyzers, dictation systems, and voice information retrieval systems.

Dr. Lee was Vice President for International Affairs (1996–1997) and the Awards Committee chair (1998–2005) of the IEEE Communications Society. He was a member of the Board of International Speech Communication Association (ISCA 2002–2009), a Distinguished Lecture (2007-2008) and a member of the Overview Paper Editorial Board (since 2009) of the IEEE Signal Processing Society. He is a fellow of ISCA since 2010, and received the Meritorious Service Award from IEEE Signal Processing Society in 2011.