Parallel Absolute-Relative Feature Based Phonotactic Language Recognition

Weiwei Liu, Wei-Qiang Zhang, Zhiyi Li, Jia Liu

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
liu-wwl10@mails.tsinghua.edu.cn

1. Introduction

Language recognition is the process of identifying a language from an utterance. It is an essential technology used in many applications, such as speech translation, multilingual speech recognition and information security and forensics [1]. Currently, acoustic systems [2] and phonotactic systems [1] are two types of language recognition systems widely used.

In these two types of language recognition systems, the process of feature extraction of an utterance is independent of other utterances. Such feature is a kind of "absolute feature". Focusing on "what it is", absolute feature describes an utterance directly, e.g. the phoneme strings, the probability of a sequence in the lattice, Mel-frequency cepstrum (MFCC), etc.

In this paper, we propose a new feature that is based on the relationship between an utterance and a set of selected datum utterances, so called "relative feature". Relative feature cares about "where it is" instead of "what it is". Then an utterance is positioned in the relative coordinate. By showing the relationship of the utterances directly, relative feature makes utterances more convenient to classify. The relationship can be distance, similarity, angle, etc. Relative feature varies with the different relationship measurement. By selecting a proper measurement we can strengthening identifiable character or discard nuisance attribute of the utterance.

In language recognition systems, the most identifiable character is the similarity between training and test data. So here we discuss a relative feature defined as the similarity between the feature supervectors of utterances. There are many ways to discriminate between utterances represented by similarities. Nearest neighbor (NN) is one of the most traditional methods [3]. However, it suffers from the limitation of high-computational complexity, sensitivity to noise. The widely used kernel method can be thought of as a nonlinear similarity measure. Kernel method is an efficient similarity measurement between two supervectors, which is introduced into face recognition, handwritten digit recognition [4] to achieve higher robustness to noise [5]. In this paper, we will introduce a kind of kernel called empirical kernel [6] into language recognition and develop a relative feature. Using the feature supervectors that are already built in language recognition system we can easily compose the new feature with only a little more inner product computation.

Using the "absolute feature" of all the training data and "relative feature" of a small subset of training dataset, a parallel absolute-relative feature (PARF) language recognition system is built and achieves good performances for LRE tasks.

The remainder of this paper is organized as follows: Section 2 introduces the traditional language recognition system. Section 3 shows the difference in formulation between the absolute-feature-based phonetic language recognition system and the relative-feature-based language recognition system, which is constructed in the relative feature kernel space. Experiments for evaluating the proposed approach are presented in Section 5. Finally, Section 6 concludes this paper.

2. Baseline System

In this work a PR-SVM [7] phonotactic language recognition system is used as baseline system. Basically, the traditional language recognition system works by mapping the input data space X into a high dimensional feature space F: \( \Phi: X \rightarrow F \), and then building linear machines in the feature space to implement their nonlinear counterparts in the input space to classify.

In PR-SVM language recognition systems, an utterance can be mapped to the feature space as follows:

\[
\Phi: x \rightarrow \varphi(x)
\]

Then the feature supervector \( \varphi(x) \) is sent to the classifier and a decision is made based on the output of the classifier. In phonotactic language recognition system [8],

\[
\varphi(x) = [p(d_1|\ell_x), p(d_2|\ell_x), ..., p(d_F|\ell_x)],
\]

here \( d_i \) is the n-gram phoneme string \( d_i = s_{i}...s_{i+n-1} \) (n = N) and \( F = f^N \) (f is the size of the phone inventory for a single phone recognizer and N is the number of n-gram). \( \ell_x \) is the lattice converted from data x by a phone recognizer. \( p(d_i|\ell_x) \) denotes the probability of the N-gram \( s_{i}...s_{i+N-1} \) in the lattice.

\( s_{i} \)
If we employ a SVM as the classifier, a decision is based on the SVM output score compared to a threshold as follows:

\[ f(\varphi(x)) = \sum_{i} \alpha_{i}K_{\text{TFLLR}}(\varphi(x), \varphi(x_{i})) + d, \tag{3} \]

Here \( \varphi(x_{i}) \) are support vectors obtained from training set using the Mercer condition. \( K_{\text{TFLLR}} \) is a term frequency log-likelihood ratio kernel which is computed as [9]:

\[ K_{\text{TFLLR}}(\varphi(x_{i}), \varphi(x_{j})) = \frac{p(d_{i}|l_{x_{j}})}{\sqrt{p(d_{i}|l_{x})}} \times \frac{p(d_{j}|l_{x})}{\sqrt{p(d_{j}|l_{x})}}. \tag{4} \]

The \( p(d_{i}|l_{x_{j}}) \) is calculated from the observed probability across all lattices. In language recognition system the training is always carried out with a one-versus-rest strategy. The samples in the target language are viewed as the positive set and the remainder as the negative one. Then the training stage is carried out between the positive set and negative set. 

3. Parallel Absolute-Relative Feature Followed by SVM System

3.1. Relative feature kernel map

The novelty of relative feature kernel map lies in interpreting kernel here as a representation of a similarity space. To construct a relative feature kernel map, a data set \( S \) of size \( m \) will be used as the datum mark of similarity. Here \( S = [s_{1}, s_{2}, ..., s_{m}] \) is a datum set that is selected from training corpus by VQ method, which selects the set of utterance that could present their type of language most. \( S \) is mapped into feature space like this:

\[ S \rightarrow \varphi(S) = [\varphi(s_{1}), \varphi(s_{2}), ..., \varphi(s_{m})] \tag{5} \]

The relative feature kernel between two supervector \( \varphi(x_{i}) \) and \( \varphi(x_{j}) \) is

\[ K_{\text{RF}}(\varphi(x_{i}), \varphi(x_{j})) = <\varphi(x_{i}), \varphi(x_{j})> = \sum_{q=1}^{p} \frac{p(d_{i}|l_{x_{j}})}{\sqrt{p(d_{i}|l_{x})}} \times \frac{p(d_{j}|l_{x})}{\sqrt{p(d_{j}|l_{x})}} \tag{6} \]

Relative feature kernel is similar to TFLLR kernel except normalized by the observed probability across all lattices of datum dataset \( p(d_{i}|l_{x}) \). Relative feature kernel indicates that a high degree of similarity will exist between the two supervectors, and vice versa.

Then the utterance \( x \) is mapped from the input data space \( \mathcal{X} \) to an \( m \)-dimensional Euclidean space \( \mathbb{R}^{m} \): \( \Phi^{r}: \mathcal{X} \rightarrow \mathbb{R}^{m} \) as follows:

\[ \Phi^{r}: x \rightarrow \varphi^{r}(x) = K_{\text{RF}}(\varphi(x), \varphi(S)) = <\varphi(x), \varphi(S)> = [K_{\text{RF}}(\varphi(x), \varphi(s_{1})), ..., K_{\text{RF}}(\varphi(x), \varphi(s_{m}))] \tag{7} \]

In general, \( K_{\text{RF}}(\varphi(x), \varphi(S)) \) defines a vector consisting of \( m \) similarities found between the utterance \( x \) and all the utterance in the datum set \( S \). In particular, \( K_{\text{RF}}(\varphi(x), \varphi(S)) \) is treated as a description of a space where each dimension corresponds to the similarities to a prototype. Therefore, \( K_{\text{RF}}(\varphi(S)) \) is viewed as a mapping onto an \( m \)-dimensional empirical kernel feature space.

3.2. PARF system for language recognition

The architecture of the PARF system is shown in Fig.1. In absolute feature part, all the training data is mapped into feature space for training. An utterance \( x \) is mapped to \( \varphi^{r}(x) \) and a decision is made by eq.(3). In relative feature part, we use some small subsets of training dataset for training (each subset must contain utterances for all target languages). Then an utterance \( x \) is mapped to \( \varphi^{r}(x) \) in the relative feature space. If we employ a SVM as the classifier, a decision is based on the SVM output compared to a threshold as follows:

\[ f'(\varphi^{r}(x)) = \sum_{r} \alpha_{r}K'(\varphi^{r}(x), (\varphi^{r}(x_{r}))+d', \tag{8} \]

Here \( \varphi^{r}(x_{r}) \) are support vectors obtained from training set using the Mercer condition. When \( K' \) is RBF kernel, \( K' \) is computed as follows:

\[ K'_{\text{RBF}}(\varphi^{r}(x_{i}), \varphi^{r}(x_{j})) = \exp(-\gamma|\varphi^{r}(x_{i}) - \varphi^{r}(x_{j})|^{2}), \tag{9} \]

Here \( \gamma = 1/\lambda \), \( \lambda \) is the dimension of the relative feature. And if \( K' \) is adopted as TFLLR kernel, \( K' \) is computed as follows:

\[ K'_{\text{TFLLR}}(\varphi^{r}(x_{i}), \varphi^{r}(x_{j})) = \sum_{q=1}^{m} \frac{K_{\text{RF}}(\varphi(x_{i}), \varphi(s_{q})) + K_{\text{RF}}(\varphi(x_{j}), \varphi(s_{q}))}{K_{\text{RF}}(\varphi(x_{i}), \varphi(s_{q}))} \tag{10} \]

In PARF language recognition system, the training is also carried out with a one-versus-rest strategy. The samples in the target language are viewed as the positive set and the remainder as the negative one. Then the training stage is carried out between the positive set and negative set.

Then the LDA-MMI method is used to maximize the posterior probabilities of all the belief score vector [10] with objective function as follows [11]:

\[ F_{\text{MMI}}(\lambda) = \log \sum_{i} \frac{P(\varphi(x_{i}|\lambda_{g(i)})P(g(i))}{\sum_{j} P(\varphi(x_{j}|\lambda_{j})P(j))} \tag{11} \]

Here \( x = [w_{f}(\varphi(x)), w_{1}f'(\varphi^{r}(x)), ..., w_{s}f'_{s}(\varphi^{r}(x)) \) and \( g(i) \) indicates its class label. \( w, w_{1}, w_{2}, ..., w_{s} \) are weights of the belief of the absolute feature and relative feature. \( \sum_{i} \omega_{w} + w = 1 \). Usually \( \omega = M/(M + \sum_{i} M_{\text{sub}}) \), \( \omega_{m} = (M_{\text{sub}}/M + \sum_{i} M_{\text{sub}}) \). \( M \) denotes the number of the training utterances and \( M_{\text{sub}} \) denotes the number of the subset of training utterances that used to produce the relative feature. \( P(j) \) is the prior probability of class \( j \). \( p(\varphi(x)|\lambda) \) is weighted Gaussian mixtures that describe a general distribution:

\[ p(\varphi(x)|\lambda) = \sum_{q=1}^{m} \omega_{m}^{q}N(\varphi(x; \mu_{m}, \Sigma_{m})), \tag{12} \]

Here \( N(\cdot) \) denotes the normal distribution with a parameter set that is often referred as \( \lambda = \{\omega_{m}, \mu_{m}, \Sigma_{m}\} \). Here \( \mu_{m}, \Sigma_{m} \) and \( \omega_{m}' \) are the mean vector, covariance matrix and the weight of the \( m \)-th Gaussian mixture.

Such a language recognition system has three advantages. First, relative feature describes the utterance from a different aspect from the traditional feature extract method, so the whole system can extract more useful information to classify. Second, the dimension of absolute feature \( F \) is very high, while relative feature has flexible dimension with the number of datum utterance \( m \). Generally \( m < < F \). Thus the size of the
input vector of the relative feature can be smaller than absolute feature, which means less computation. Third, to improve the performance of language recognition, traditional language recognition systems employ different phone recognizers [7] and use different acoustic models [12] to provide different kinds of features to describe the utterances from different aspect. Both of the two methods need to build new features again and again, so both of them multiply the computation for times. While the process of building relative feature uses middle product of the traditional language recognition (eg. the absolute feature $\psi(x)$ that have already built), what we need to compute is only $\psi(S)$ and $K(\cdot, \psi(S))$, which cost a little computation. Thus the PARF language recognition system can build two features from different aspects with computing just once.

4. Experimental setup

4.1. Baseline language recognition system

In this work a PR-SVM phonotactic language recognition system is used as baseline system. First, the Temporal Patterns Neural Network (TRAPs/NN) phone decoders developed by the Brno University of Technology (BUT) for Hungarian (HU) [13] are applied to compute phone posteriori probability. Then, the HVite decoder [14] produced by HTK is used to produce phone lattices. Then, a popular SVMTorch [15] from SVM package is used to give SVM scores. Finally, the LDA-MMI algorithm [16] is used for score calibration.

4.2. Test, training, developing and datum dataset

The experiments are conducted on the test trials of the 2009 NIST Language Recognition Evaluation (NIST-LRE2009) tasks in which 41793 test segments are involved in for 30-s, 10-s, and 3-s nominal duration test. There are totally 23 target languages, including Amharic, Bosnian, Cantonese, Creole (Haitian), Croatian, Dari, English (American), English (Indian), Farsi, French, Georgian, Hausa, Hindi, Korean, Mandarin, Pashto, Portuguese, Russian, Spanish, Turkish, Ukrainian, Urdu, and Vietnamese. And the evaluation involves Voice of America (VOA) radio broadcasts and conversational telephone speech (CTS) channel conditions.

Train data used in the experiment are (1) the Call-Home Corpus; (2) the Call-Friend Corpus; (3) the OGI Corpus; (4) the OHSU Corpus provided by NIST for LRE05; and (5) the VOA Corpus. 22701 conversations are selected from the dataset provided by NIST for the 2003, 2005 and 2007 LRE and VOA are used for develop purposes.

Almost 14000 conversations which are selected from 40 languages by VQ method provided by NIST for the 2003, 2005 and 2007 LRE and VOA are used as datum dataset.

4.3. Evaluation measures

In NIST-LRE2009, the performance of language recognition systems is reported in terms of: (1) Equal Error Rate (EER); (2) average cost performance $C_{avg}$ as defined by NIST [17].

5. Experimental Results and Discussion

In this work a PR-SVM [7] phonotactic language recognition system is used as baseline system. Here 1804213 systems were applied to compute phone posteriori probability. Then, the HVite decoder [14] produced by HTK is used to produce phone lattices. Then, a popular SVMTorch [15] from SVM package is used to give SVM scores. Finally, the LDA-MMI algorithm [16] is used for score calibration.

Table 1: Performance of baseline system and PARF system (K = RBF kernel). LRE 09, HU frontend (EER/Cavg in %). PARF-$n$ means the number of the datum utterances taking part in the process of building relative feature.

<table>
<thead>
<tr>
<th></th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>2.17/1.98</td>
<td>7.61/7.54</td>
<td>23.90/23.42</td>
</tr>
<tr>
<td>PARF-1000</td>
<td>1.96/1.86</td>
<td>6.28/6.20</td>
<td>21.17/21.15</td>
</tr>
<tr>
<td>PARF-3000</td>
<td>1.94/1.87</td>
<td>6.21/6.16</td>
<td>20.89/20.97</td>
</tr>
<tr>
<td>PARF-5000</td>
<td>1.92/1.82</td>
<td>6.15/6.10</td>
<td>20.72/20.49</td>
</tr>
<tr>
<td>PARF-7000</td>
<td>1.87/1.76</td>
<td>6.15/6.06</td>
<td>20.71/20.30</td>
</tr>
</tbody>
</table>

Table 2: Performance of baseline system and PARF system (K = TFLLR kernel). LRE 09, HU frontend (EER/Cavg in %). PARF-$n$ means the number of the datum utterances taking part in the process of building relative feature.

<table>
<thead>
<tr>
<th></th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>2.17/1.98</td>
<td>7.61/7.54</td>
<td>23.90/23.42</td>
</tr>
<tr>
<td>PARF-1000</td>
<td>1.98/1.87</td>
<td>6.37/6.23</td>
<td>20.87/20.74</td>
</tr>
<tr>
<td>PARF-2000</td>
<td>1.97/1.84</td>
<td>6.35/6.23</td>
<td>20.60/20.46</td>
</tr>
<tr>
<td>PARF-3000</td>
<td>1.92/1.81</td>
<td>6.20/6.21</td>
<td>20.45/20.40</td>
</tr>
<tr>
<td>PARF-4000</td>
<td>1.92/1.83</td>
<td>6.20/6.11</td>
<td>20.41/20.57</td>
</tr>
<tr>
<td>PARF-5000</td>
<td>1.90/1.81</td>
<td>6.10/6.08</td>
<td>20.02/20.04</td>
</tr>
</tbody>
</table>
Table 3: Performance of baseline system and PARF system (K’= RBF and TFLLR kernel). LRE 09, HU frontend (EER/Cavg in %).

<table>
<thead>
<tr>
<th></th>
<th>30s</th>
<th>10s</th>
<th>3s</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>2.17/1.98</td>
<td>7.61/7.54</td>
<td>23.90/23.42</td>
</tr>
<tr>
<td>PARF</td>
<td>1.84/1.75</td>
<td>6.04/5.99</td>
<td>19.89/19.67</td>
</tr>
</tbody>
</table>

6. Conclusion

We have presented in this paper an approach to build “relative features” and a parallel absolute-relative feature language recognition system. We use the measurement of relationship between different utterances to build relative feature, which can describe an utterance more efficiently. The relative feature kernel map provides a method that can describe the similarity to the datum set for both training and test utterance. Our experiments confirm that the PARF system leads to an improvement in the performance of language recognition system. Experiments reveal that it does increase the performance of the language recognition system on accuracy without sacrificing its structure simplicity and computational effort much. In the future researches, we would like to study the way of efficient datum utterance selection, kernel diversification of kernel feature map for language recognition.

7. Acknowledgements

This project is supported by National Natural Science Foundation of China (No. 61005019 and No.61273268).

8. References


