Subspace-Based Feature Representation and Learning for Language Recognition

Yu-Chin Shih[1, 2], *Hung-Shin Lee*[1, 2], *Hsin-Min Wang*[2], *Shyh-Kang Jeng*[1]

[1]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[2]Institute of Information Science, Academia Sinica, Taipei, Taiwan
{ycshih, hslee, whm}@iis.sinica.edu.tw, skjeng@ew.ee.ntu.edu.tw

Abstract

This paper presents a novel subspace-based approach for phonotactic language recognition. The whole framework is divided into two parts: the speech feature representation and the subspace-based learning algorithm. First, the phonetic information as well as the contextual relationship, possessed by spoken utterances, are more abundantly retrieved by likelihood computation and feature concatenation through the decoding processed by an automatic speech recognizer. It is assumed that the extracted phone frames reside in a lower dimensional eigen-subspace, in which the structure of data can be approximately captured. Each utterance is further represented by a fixed-dimensional linear subspace. Second, to measure the similarity between two utterances, suitable non-Euclidean metrics are explored and applied to non-linear discriminant analysis in a kernel fashion, followed by a back-end classifier, such as the *k*-nearest neighbor (K-NN) classifier. The results of experiments on the OGI-TS database demonstrate that the proposed framework outperforms the well-known vector space modeling based method with relative reductions of 38.90% and 27.13% on the 1-to-50-second and 3-second data sets respectively in equal error rate (EER).

Index Terms: language recognition, subspace-based learning

1. Introduction

Typically, automatic spoken language recognition (ASLR) techniques fall into two major categories according to how spoken utterances are tokenized for back-end classifiers [1]. In the acoustic aspect, a series of frames, each of which contains 80-200 ms temporal information, such as MFCC or SDC [2], is directly derived from raw speech by speech parameterization processes. In contrast, phonotactic approaches exploit phone recognizers to convert acoustic frames of an utterance into a sequence of phone symbols to capture longer-term information. Phonotactic constraints across languages, e.g., permissible syllable structures or phone combinations, are considered in this aspect so that the characteristics of a language can be well modeled [3].

Going into details, various implementations to deal with phonotactic features have been proposed, from the use of several *n*-gram language model-based phonetic decoders for each single language (PPR) [4], the use of a single phonetic decoder followed by the computation of language dependent phone *n*-gram likelihoods (PRLM) [5], to the use of paralleled single-language phonetic decoders followed by a phone *n*-gram classifier (PPRLM) [6]. However, the speech data in the above three approaches is processed through several independent channels composed of front-end phonetic decoders and back-end language models, and output information is fused only at the final score level. To model the correlation among different phonetic decoders, vector space modeling (VSM) and support vector machines (SVM) are introduced. Stemming from the well-known VSM framework in the field of information retrieval (IR), Li *et al.* proposed to build a composite feature vector for each utterance by concatenating the vector-formed statistics corresponding to the phonetic decoders, and to apply SVM to composite vectors for classification [7]. In their work, each of phone or sound sequences can be represented by a high dimensional phonotactic feature vector with *n*-gram counts or the term frequency-inverse document frequency (TF-IDF), whose dimensionality is equal to the total number of phonotactic patterns needed to characterize the structure of the utterance given by one decoder. Moreover, M. Penagarikano *et al.* took time alignment information into account by considering time-synchronous cross-decoder phone co-occurrences [8]. They have thus defined a new concept of multi-phone labels, which attempts to integrate the contributions given by several decoders frame by frame and forms a VSM-based label sequence different from the conventional *n*-gram patterns.

From the viewpoint of data representation, VSM has merits to well serve the back-end learning mechanism like SVM or GMM by transforming each varying-length phone sequence into a fixed-length vector. Furthermore, to alleviate the order-losing problem that the order in which the phones appear in an utterance is lost in the VSM representation, and to relax the assumption that phones are statistically independent to some extent, more phonotactic attributes, such as bigrams and trigrams, are included to form a much higher dimensional vector. In this paper, we propose a new approach for data representation. In this approach, the phonetic information as well as the contextual relationship can be more abundantly retrieved by likelihood computation and feature concatenation, given a universal phone recognizer. In the VSM framework, the count or frequency of a phonotactic term is the only attribute that might be concerned. Additionally, since the total number of *n*-gram patterns tends to increase exponentially with respect to *n*, *n* is often inevitably limited to 2 (bigram) due to the capacity of memory, or to 3 (trigram) but with some term-size reduction. In contrast, our approach enables us to look at much farther and deeper through the decoder’s eyes. That is to say, not only can more information, like posterior probabilities or likelihood scores of any phone segments, be captured, but also all possible phones can be taken into account instead of just using the most likely one. The spirit is similar to the employment of phone lattices on phonotactic language recognition [9]. Our representation also fits for the back-end classification. Under the assumption that the representation can be approximately modeled by a collection of lower dimensional linear subspaces, a suitable subspace-based learning algorithm along with the well-surveyed Grassmann kernel can be introduced for classification, even while the size of subspaces for each utterance varies [10].

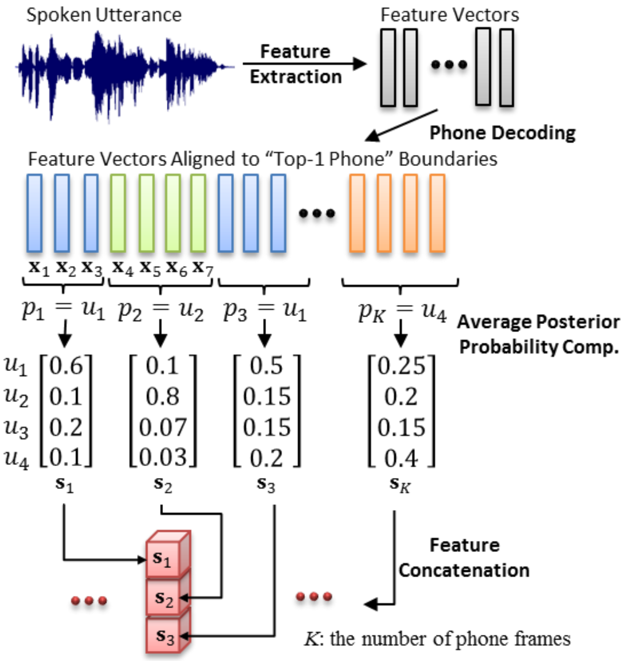
Spoken Utterance
Feature Extraction
Feature Vectors
Phone Decoding
Feature Vectors Aligned to "Top-1 Phone" Boundaries
$\mathbf{x}_1$ $\mathbf{x}_2$ $\mathbf{x}_3$ $\mathbf{x}_4$ $\mathbf{x}_5$ $\mathbf{x}_6$ $\mathbf{x}_7$
$p_1 = u_1$ $p_2 = u_2$ $p_3 = u_1$ $p_K = u_4$
Average Posterior Probability Comp.
u_1 u_2 u_3 u_4
0.6 0.1 0.2 0.1
0.1 0.8 0.07 0.03
0.5 0.15 0.15 0.2
0.25 0.2 0.15 0.4
$\mathbf{s}_1$ $\mathbf{s}_2$ $\mathbf{s}_3$ $\mathbf{s}_K$
Feature Concatenation
$\mathbf{s}_1$ $\mathbf{s}_2$ $\mathbf{s}_3$
K: the number of phone frames

Figure 1: *Phonotactic feature extraction and frame concatenation for a four-phone $\{u_1, u_2, u_3, u_4\}$ ASLR system.*

The rest of the paper is organized as follows. In Section 2, we introduce the new representation of a spoken utterance in the sense of subspaces. In Section 3, we present the learning mechanism for subspaces with Grassmann kernels. Section 4 gives the evaluation results and some discussions. Finally, conclusions and future work are outlined in Section 5.

2. Data representations and subspaces

2.1. Phonotactic feature extraction

Given the observed sequence of acoustic vectors $\mathbf{x}_1, \ldots, \mathbf{x}_T$ derived from a spoken utterance, phonetic decoders single out the best phone sequence $p_1, \ldots, p_K$ based on Viterbi approximation, which finds the most likely time alignment path through a huge probabilistic network. The main task of phonotactic-based language classifiers is to take advantage of the phone sequence $p_1, \ldots, p_K$ as a basic unit for language identification or verification. The basic idea behind our proposed phonotactic data representation is to take the single-best phone sequence produced by phonetic decoders as a kind of clusters toward the acoustic feature vectors in a phonotactic fashion. From the example in Figure 1, we can see that after phone decoding and time alignment, feature vectors $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$ belong to top-1phone u_1 while feature vectors $\{\mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7\}$ belong to top-1 phone u_2. Along this vein, given phone boundaries, each set of acoustic vectors in its corresponding phone segment is further used to derive a more meaningful phonotactic feature vector built up with the average posterior probability for each phone, which is characterized by a hidden Markov model (HMM), through the Viterbi search algorithm. Consequently, each phone segment (or phone frame) $\mathbf{s}_k$ is expressed by a vector, whose dimensionality is the size of the universal phone set $\{u_i\}$. Figure 1 also shows that the phone frame p_1, which is labeled as u_1, indeed has the highest posterior probability of 0.6 with respect to u_1 due to the nature of dynamic programming contributed by the first three feature vectors $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$. As for u_2, u_3, and u_4 in this phone frame, although the posterior probabilities might be much smaller than the single-best u_1, and even u_3 never appears in the phone sequence, they are not cast off but included into the new phonotactic feature vector $\mathbf{s}_1$ to bring more uncertainty or information that might be helpful for classification.

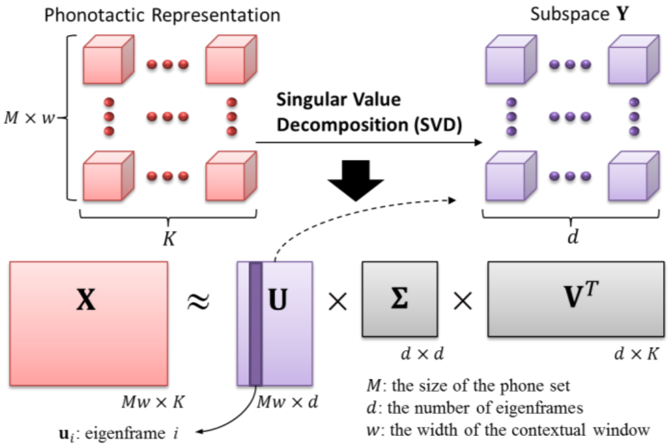
Phonotactic Representation
Subspace $\mathbf{Y}$
$M \times w$
Singular Value Decomposition (SVD)
K
d
$\mathbf{X} \approx \mathbf{U} \times \boldsymbol{\Sigma} \times \mathbf{V}^T$
$Mw \times K$
$Mw \times d$
$d \times d$
$d \times K$
$\mathbf{u}_i$: eigenframe i
M: the size of the phone set
d: the number of eigenframes
w: the width of the contextual window

Figure 2: *Subspace generation by SVD.*

2.2. Frame concatenation

In order to capture the contextual information as well as to learn patterns of phonotactic constraints, like the role that *n*-gram terms play in the VSM scheme, we concatenate the phone frames belonging to a given contextual window centered on a current phone frame $\mathbf{s}_k$. Let $w = 2n + 1$ denotes the width of the contextual window, meaning that the larger the value *n*, the more phonotactic patterns can be modeled. In Figure 1, if we set *n* to 1, the 3 consecutive phone frames, $\mathbf{s}_1$, $\mathbf{s}_2$, and $\mathbf{s}_3$, are spliced together forming a context-dependent supervector. Moreover, it is worth noting that the size of the phonotactic representation no longer increases exponentially with respect to *n*; on the contrary, the size has a linear growth that guarantees to avoid memory deficiency and unnecessary reduction in practice.

2.3. Subspace generation

After frame concatenation, each utterance can be expressed by a matrix $\mathbf{X}$ consisting of $(M * w)$ rows and K columns, where M and K denote the size of the phone set and the number of phone frames in an utterance, respectively. Since K varies in utterances, we employ singular value decomposition (SVD) to map the varying-size phonotactic matrix to a fixed-size pattern as depicted in Figure 2 [10]. The SVD of $\mathbf{X}$ can be written as a reduced form, $\mathbf{X} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, where $\boldsymbol{\Sigma}$ is a $d \times d$ diagonal matrix containing the largest d singular values $\sigma_1, \ldots, \sigma_d$, $\mathbf{U}$ and $\mathbf{V}$ are matrices, whose orthonormal columns $\{\mathbf{u}_i\}$ and $\{\mathbf{v}_i\}$ approximately span the column and row spaces of $\mathbf{X}$, respectively. Our goal here is not to find a lower-dimensional projection of $\mathbf{X}$, but to find a linear subspace that can approximately span the whole column spaces of $\mathbf{X}$. In other words, what we need is the columns $\{\mathbf{u}_i\}$ of $\mathbf{U}$, called eigenframes, that can characterize each phone frame in $\mathbf{X}$. Therefore, each utterance j is represented as $\mathbf{Y}_j = [\mathbf{u}_1, \ldots, \mathbf{u}_d]$, a collection of orthonormal bases belonging to a fixed-dimensional linear subspace of a Euclidean space. The main difference between our representation and other subspace-based methods for an utterance lies in that, we represent an utterance as

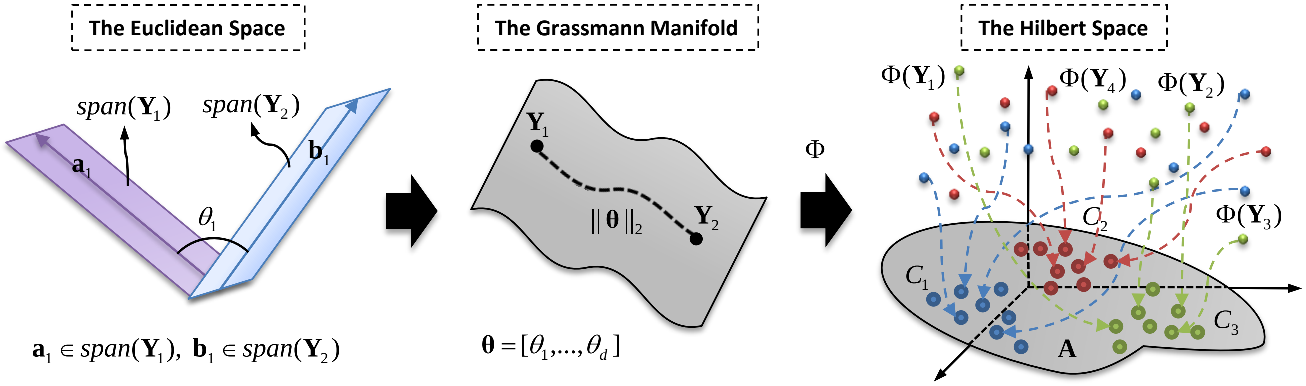

Figure 3: *Subspace learning and kernel linear discriminant analysis.*

a *linear space* with a set of bases while other subspace-based methods, such as iVector [16] and principal component analysis (PCA) [17], attempt to derive a *coordinate representation* (or location) for an utterance in a linear space where all utterances share the same set of bases. The next question is how to apply the representation to a classification task. That is, a suitable distance measure and a subspace-based learning mechanism have to be explored.

3. Subspace learning

3.1. Grassmann metric

As described in the previous section, $\mathbf{Y}_j$ is a matrix of size D by d with orthonormal columns, where $D = M * w$. The Grassmann manifold can be defined as the set of $\mathbf{Y}_j$'s, where 1) any two subspaces $span(\mathbf{Y}_1)$ and $span(\mathbf{Y}_2)$ can be treated as two points, whose Riemannian distance is computed by $d(\mathbf{Y}_1, \mathbf{Y}_2) = \|\boldsymbol{\theta}\|_2$ in terms of the d principal angles, as shown in Figure 3; 2) $\mathbf{Y}_1$ and $\mathbf{Y}_2$ is equivalent if and only if $span(\mathbf{Y}_1) = span(\mathbf{Y}_2)$ [10]. In the literature, the principal angles between linear subspaces are a reasonable and widely-used distance measure between subspaces. The cosine of the k-th principal angle between $span(\mathbf{Y}_1)$ and $span(\mathbf{Y}_2)$, also known as the k-th canonical correlation [11], is defined by $\cos\theta_k = \max_{\mathbf{a}_k \in span(\mathbf{Y}_1)} \max_{\mathbf{b}_k \in span(\mathbf{Y}_2)} (\mathbf{a}_k \cdot \mathbf{b}_k)$ recursively, subject to $\|\mathbf{a}_k\| = \|\mathbf{b}_k\| = 1$ and $\mathbf{a}_k \cdot \mathbf{a}_i = \mathbf{b}_k \cdot \mathbf{b}_i = 0$ $(i = 1, \ldots, k-1, k \leq d)$. The d largest cosines (corresponding to the d principal angles) seem very difficult to derive. However, since $\mathbf{Y}_1$ and $\mathbf{Y}_2$ both have orthonormal columns, the cosines of all principal angles between $span(\mathbf{Y}_1)$ and $span(\mathbf{Y}_2)$ can be easily derived by the SVD of $\mathbf{Y}_1^T\mathbf{Y}_2$, expressed by $\mathbf{Y}_1^T\mathbf{Y}_2 = \mathbf{U}(\cos\boldsymbol{\Theta})\mathbf{V}^T$, where $\cos\boldsymbol{\Theta} = diag(\cos\theta_1, \ldots, \cos\theta_d)$ [11].

Many metrics can be used to measure the distance between subspaces based on the principal angles. In this paper, we adopt the Projection metric as a distance measure between two subspaces $\mathbf{Y}_i$ and $\mathbf{Y}_j$ of two utterances, which is defined by $d(\mathbf{Y}_i, \mathbf{Y}_j) = \left(d - \sum_{k=1}^{d} \cos^2\theta_k\right)$ [10]. The Projection metric has been also proved to be induced from a positive definite kernel that makes kernel-based algorithms, such as SVM and kernel linear discriminant analysis (KLDA), efficiently solvable [12].

3.2. Kernel LDA and the nearest neighbor classifier

The basic idea in linear discriminant analysis (LDA) is to find a lower dimensional projection of the data corresponding to the m generalized eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$ associated with the m largest generalized eigenvalues, where $\mathbf{S}_w$ and $\mathbf{S}_b$ denote the within-class and between-class covariance matrices, respectively [10]. However, as illustrated in Figure 3, due to the severe non-linearity in the Grassmann manifold, it is difficult to directly derive the discriminating features among classes in the original input space. By defining a non-linear mapping Φ that maps a subspace from the Grassmann manifold to a high-dimensional Hilbert space $R^{D \times D}$, we expect to obtain a linearly separable distribution in the Grassmann manifold [12]. Then LDA, the linear technique, can be performed to find the projective space **A** and to extract the most significant discriminating features. By introducing a suitable kernel function, which corresponds to the non-linear mapping, all the computation for KLDA can conveniently be carried out without the need to compute the mapping explicitly. The more detailed procedures and theoretic basis of KLDA using the Projection kernel can be referred to [10].

For classification, the nearest neighbor classifier is widely adopted owing to its simplicity in implementation and training. To train the classifier, one only needs to store the training instances and their class labels. In the test phase of our language verification task, given a feature vector **x** from a test utterance resulting from the subspace representation and KLDA, the probability that **x** is generated by a target language C_k can be calculated by $p(\mathbf{x}|C_k) = (1/N_k)\sum_{n=1}^{N_k} exp(-\|\mathbf{x} - \mathbf{x}_{kn}\|/2\sigma)$, where σ is a scaling factor, $\{\mathbf{x}_{kn}|n = 1, \ldots, N_k\}$ are the feature vectors of N_k training samples belonging to the target language C_k [13]. The probability can be taken as a score for likelihood ratio computation used in spoken language verification.

4. Experiment results and discussion

We conducted the language verification task on the Oregon Graduate Institute Multi-language Telephone Speech (OGI-TS) Corpus [14], which contains the speech of 10 languages: English, Farsi, French, German, Japanese, Korean, Mandarin, Spanish, Tamil, and Vietnamese. The corpus is divided into three parts: 4650 utterances for training, 1899 utterances for development, and 1848 utterances for test. Some test utterances with lengths ranging from 2 to 4 seconds are culled to form the 3-s set to evaluate the system performance for short utterances, while all test utterances form the 1-to-50-s set. Besides, the corpus also includes 619 "story-before-tone" utterances, which have hand generated fine-phonetic transcriptions that can be used for supervised phone modeling for six languages.

A universal phone recognizer, which is composed of a set of language-independent context-independent phone models, is

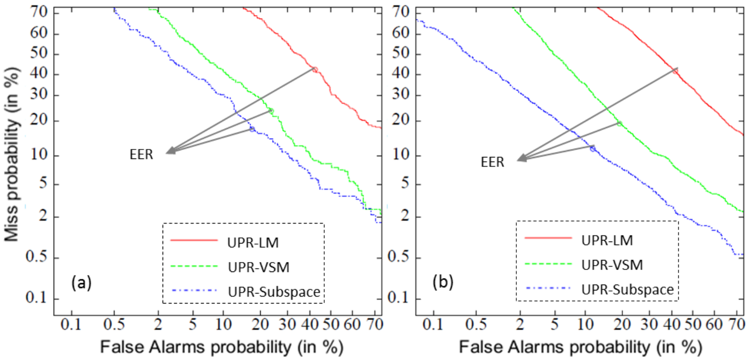

Figure 4: *DET plots for several phonotactic approaches on (a) the 3-s data set and (b) the 1-to-50-s data set.*

used in the experiments. Each phone model models a phone in the universal phone inventory of size 69, which is a union of the phones appearing in the 619 transcription files, with the phones sharing the same manner and place merged into one. Each phone model is a 3-state left-to-right CD-HMM with 32 Gaussian mixture components per state. The acoustic feature vector has 39 attributes comprised of 13 MFCCs including C0, along with their first and second order derivatives. All phone models were trained and refined from the 619 phone-transcribed utterances and the 4650 training utterances according to the maximum likelihood criterion, respectively.

We compared the proposed method with two well-known phonotactic language recognition methods, namely UPR-LM and UPR-VSM [7]. UPR-LM is an extension of PRLM by using a universal phone recognizer instead of a phone recognizer of only one specific language. We applied the bigram back-off language model with Good-Turing smoothing in the implementation. For UPR-VSM, a phone sequence is represented by a $(69 + 69^2)$ dimensional vector consisting of the TF-IDF of unigram and bigram phonotactic patterns. Latent semantic indexing (LSI) was further used for extracting key features needed for discriminating spoken utterances from the statistics of some salient units and their co-occurrence. In the back-end classification, UPR-VSM used the SVM system with an RBF kernel trained for each target language individually. The decision for each utterance was made based on the posterior probability given by applying the SVM output to a sigmoid function. Since the proposed method is also based on a universal phone recognizer front-end, it is denoted as UPR-Subspace.

From Figure 4, we can see that UPR-subspace outperforms UPR-LM and UPR-VSM. UPR-subspace achieves relative reductions of 38.90% and 27.13% over UPR-VSM in equal error rate (ERR) on the 3-s and 1-to-50-s data sets, respectively. Figure 5 (a) shows EER with respect to the number of eigen-frames (d). The results demonstrate that, with a moderate value of d, the EER is dramatically degraded. From Figure 5 (b), we observe that larger widths $(n > 1, w > 3)$ of the contextual window not necessarily guarantee lower EERs. We can also see that the ERR is minimal when $n = 1$ $(w = 3)$. The results seem to somewhat explain the reason why the maximum size of phonotactic constraints are set to 3 (trigram) in some phonotactic learning simulation systems [15] from another viewpoint.

5. Conclusion

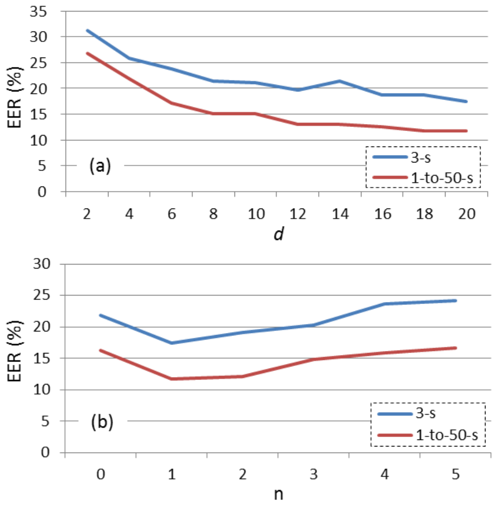

Figure 5: *EER with respect to (a) d, the number of eigenframes, and (b) n, the width of the contextual window* $(w = 2n + 1)$.

This paper presents a new phonotactic feature representation based on subspace formulation for automatic spoken language recognition. On the basis of the representation, the combination of KLDA and the simple nearest neighbor classifier has been shown to perform excellently. It is expected that the integration of a more discriminative kernel-based classier, such as SVM, with a subspace-based kernel could perform even better. In our future work, we plan to evaluate the proposed framework on the more standard NIST LRE corpora, and other subspace-based methods will be implemented and compared in experiments.

6. References

[1] E. Ambikairajah *et al.*, "Language identification: a tutorial," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82-108, 2011.

[2] P. A. Torres-Carassquilo *et al.*, "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *Proc. ICSLP*, 2002.

[3] A. S. House and E. P. Neuburg, "Toward automatic identification of the language of an utterance. I. Preliminary methodological considerations," *J. Acoust. Soc. Amer.*, vol. 62, no. 3, pp. 708-713, 1977.

[4] Y. K. Muthusamy *et al.* "A comparison of approaches to automatic language identification using telephone speech," in *Proc. Eurospeech*, 1993.

[5] T. J. Hazen and V. W. Zue, "Segment-based automatic language identification," *J. Acoust. Soc. Amer.*, vol. 101, no. 4, pp. 2323-2331, 1997.

[6] M. A. Zissman and E. Singer, "Automatic language identification of telephone speech messages using phoneme recognition and N-gram modeling," in *Proc. ICASSP*, 1994.

[7] H. Li *et al.*, "A vector space modeling approach to spoken language identification," *IEEE Trans. Audio, Speech, Lang. Process*, vol. 15, no. 1, 2007.

[8] M. Penagarikano *et al.*, "Improved modeling of cross-decoder phone co-occurrences in SVM-based phonotactic language recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2348-2363, 2011.

[9] J. L. Gauvain *et al.*, "Language recognition using phone lattices," in *Proc. ICSLP*, 2004.

[10] J. Hamm, and D. D. Lee, "Grassmann discriminant analysis: a unifying view on subspace-based learning," in *Proc. ICML*, 2008.

[11] A. Björck and G. H. Golub, "Numerical methods for computing angles between linear subspaces," *Math. Computation*, vol. 27, no. 123, 1973.

[12] J. Hamm. *Subspace-Based Learning with Grassmann Kernels*, PhD Thesis, 2008.

[13] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, New York, 1986.

[14] A. Y. K. Muthusamy *et al.*, "The OGI multi-language telephone speech corpus," in *Proc. ICSLP*, 1992.

[15] B. Hayes and C. Wilson, "A maximum entropy model of phonotactics and phonotactic learning," *Linguistic Inquiry*, vol. 39, no. 3, pp. 379-440, 2008.

[16] M. Soufifar *et al.*, "iVector approach to phonotactic language recognition," Proc. Interspeech, pp. 2913-2916, 2011.

[17] T. Mikolov *et al.*, "PCA-based feature extraction for phonotactic language recognition," Proc. Odyssey, pp. 251-255, 2010.