International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017) pp. 8026-8033

Speaker Recognition System for Limited Speech Data Using High-Level Speaker Specific Features and Support Vector Machines

Satyanand Singh[1]
Assistant Professor, Department of Electrical and Electronics Engineering, Fiji National University, Fiji, Island.
Orcid Id: 0000-0002-7707-031X

Assaf Mansour. H[1]
Associate Professor, School of Engineering & Physics, Faculty of Science, Technology & Environment
The University of the South Pacific (USP), ICT 048 - 324, Laucala Campus, Private Mail Bag, Suva, Fiji.
Orcid Id: 0000-0003-3052-9469

Nitin Agarwal[2]
Assistant Professor, PG Department of Electronics and Communication Engineering,
Raja Balwant Singh Engineering Technical Campus, Bichpuri Agar, Uttar Paradesh, India.
Orcid Id: 0000-0003-4940-1967

Abhay Kumar[2]
Associate Professor and Head of the Department of Computer Science and Engineering,
J. B. Institute of Engineering and Technology, Hyderabad, India.
Orcid Id: 0000-0001-7327-2056

Abstract

High-level speaker-specific features (HLSSFs), such as the style of pronunciation of words, their use, phonotactics and prosody, form the main subjects of state-of-the-art research on automatic speaker recognition (ASR). In this paper, we experimentally verify HLSSF extraction and support vector machine (SVM)-based modelling techniques. The HLSSF extraction produces patterns of symbols for each speaker during ASR training. The strategy involves changing these patterns during the training and testing of ASR using frequencies (n-gram) for a given voice sample. We used SVM and n-gram frequencies to implement ASR, where the application consisted of a new kernel based on the linear log-probability proportional scoring framework. This approach yielded impressive outcomes on an assortment of abnormal state highlights in ASR. We showed that the proposed ASR based on the linear log-probability proportional scoring framework is superior to other standard log-probability frameworks. The equal error rate (EER) of our ASR method was 2.5% with a 2% improvement over the standard method.

Keywords: High-level speaker-specific features (HLSSF), Automatic Speaker recognition (ASR), Support Vector Machines (SVM), Log-Likelihood Ratio (LLR), Phone Recognition and Language Modelling (PRLM).

INTRODUCTION

Recognizing a person by his or her voice is a critical human characteristic that most of us take for granted in regular human-to-human association/correspondence. A conversation over the phone often begins by recognising one's interlocutor and, at the least in instances of recognised speakers, confirmation by the interlocutor of his/her identity in order to continue the conversation. Automatic speaker recognition (ASR) frameworks have been developed as a means of confirming identity in a number of e-trade applications, business collaboration, criminology and law enforcement. Aside from individual validation for access control, speaker recognition is a vital tool in penal issues, national security and the legal sciences. People routinely identify others by their voices with striking precision, particularly when the level of familiarity with the subject is high. Even a short, non-linguistic line for example, a quiet laugh is normally sufficient for us to recognise a well-known acquaintance [1].

In this paper, we propose an ASR system based on high-level speaker-specific features (HLSSF) and support vector machines (SVMs). SVMs [2] are two-class classifiers used to enhance speculation execution.

A key component of SVMs is that they outline elements of a high-dimensional space (SVM highlight space) and then characterize it. Although this high-dimensional space can be tricky to handle because of the number of dimensions, the SVM can deal adequately with the issue. A fundamental problem addressed in this paper is a method of representing high-level highlights using SVMs. In recent work [3], it has been claimed that one side in a conversation should be represented as a solitary vector in SVM highlight space. By using this idea, we expect to generate n-gram insights of a given conversations to be generally steady, i.e. a large part of the variety is seen in various discussion at distinctive times

due to variations in speaker sessions and the effects of the channel used. We assume that n-gram statistics within a given conversation are relatively stable, i.e. most of the variation is observed in each side of the conversation at different times, and is due to speaker session variation and channel effects. In this way, it is useful to consider one side in a given conversation as a desolate "archive" of symbols, and use one vector in SVM highlight space to archive it [4]. Therefore, it makes sense to view a side of a conversation as a single "document" of tokens and create a vector in SVM feature space for this document. A number of algorithmic and computational advances have enabled impressive ASR

performance in recent years. Approaches utilizing chiropractic data, phoneme recognisers followed by Phone Recognition and Language Modelling (PRLM) and parallel PRLM have been shown to be effective [5]. In this homeostatic structure, an arrangement of tokenism is utilized to interpret speech information as strings of symbols or cross-sections that are later scored by n-gram language models [6], or mapped into a sack of trigram highlight vectors of an SVM [7]. Scores from both kinds of system are combined and calibrated using a variety of techniques, such as Gaussian back ends [8] or multiclass logistic regression [6].

In spite of the fact that the conventional hidden Markov model (HMM)-based telephone recogniser is broadly known as among the best in class frameworks, different sorts of symbolization can be used [7] for instance, Gaussian mixture model (GMM) tokenization [8], universal phone recognition (UPR) [6], an articulator for a property-based methodology [9] and the deep neural network-based telephone recogniser [10].

PHONETIC HLSSF EXTRACTION AND CATEGORIZATION

Consider speech and extract its progression of symbols $s_1 \dots s_{t+1}$. The captured symbols might be spoken words, pitch motions and so forth. The symbols might be a discrete depiction of the data; the representation is normally important in the tokens are a discrete representation of the input speech; the representation is typically meaningful in a high-level linguistic sense.

Once the symbols have been captured, we model the speaker. The scores produced by this modelling can be used in ASR applications. While this is not the focus of this work, we briefly discuss the utility of abnormal state highlights. We use speaker-specific spoken symbols grouped using the Byblos vast vocabulary recogniser [11]. For telephonic speaker recognition, we use telephone tokenisers obtained from the Parallel Phone Recognition Language Modelling (PPRLM) framework.

The next imperative of the proposed inquiry is the method to demonstrate symbol classification. The ordinary technique for displaying symbol streams [12] can yield the probability of n-gram frequency using proportional probability, which is the proportion of the likelihood that the arrangement of the symbols was generated by an objective speaker to the likelihood that the succession was produced by utilizing n-gram circulations. Our proposed strategy uses SVMs rather than probability proportions to represent the objective speaker. We consider this methodology in the following.

Use of SVM and HLSSF in ASR

SVM System for Telephonic Speech Signals

An SVM-based telephonic ASR system scenario is as follows:

i. A person makes a call and sends a claim of identity to the ASR system.

ii. The ASR system retrieves the SVM model of the individual for all available languages.

iii. The ASR system collects the test utterance of the individual.

iv. Based on the phonetic sequence recogniser, the phonetic sequence is derived and post-processed [13].

v. The phonetic sequence is vectorised by computing frequencies of n-grams, unigrams, bigrams, term probabilities and weightings.

vi. The vector is then introduced into an SVM using the speaker's model in the appropriate language, and a score per language is generated.

vii. Language-based scores are combined based on linear weighting to produce a final score for the test utterance of the speaker.

viii. The last score is compared with a threshold, and a choice to accept or reject is made according to whether the score is above or below a threshold.

An SVM classifier separates multidimensional information acquired from two classes using a hyperplane. The model can then be utilized to foresee the class of an obscure perception relying upon its area with admiration to the hyperplane. Whenever the training speaker data are not easily recognisable, the elements can be mapped into a higher-dimensional space by using kernel capacities such that the classes can be identified [13]. A SVM is basically a "1" and "0" classifier developed from the entirety of a kernel function $K(.,.)$:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x, x_i) + b \qquad (1)$$

where y_i are the target values, $\sum_{i=1}^{N} \alpha_i y_i = 0$ and $\alpha_i > 0$, x_i are data to train the ASR. The objective values are either true or false depending on whether the comparing SV is in class "true" or "false." Based on the Mercer condition, the kernel $K(.,.)$ is represented as

$$K(x,y) = b(x)^t b(y) \quad (2)$$

Yield space $b(x)$ is ordinarily called SVM feature space.

Sequence Kernels in SVM

A test speech signal is taken and applied to SVMs intended for solo-case speech information. This part, as a rule, considers two information vectors. One approach to this issue is to utilize sequential kernels [14].

Sequential kernels examine entire groupings of information, i.e. they process $K(X,Y)$, where X and Y are sequences of speech information. Subsequent to SVM training and scoring are calculations of the kernel part, where single-occasion strategies are summed. The main point of stretching SVMs into sequential mode is that it renders them suitable for ASR in the form of kernels [15].

Kernel Construction

The appropriate selection of the kernel for SVM is the most important task in telephonic speaker recognition. It is the kernel that preserves speaker-specific information in the phone sequence. Kernel construction can be described in two steps.

The first step of kernel construction is based on a classifier that calculates the proportional probability of there being a true speaker is not going to be recognised by the ASR system [16].

The concept of kernel construction is similar to a Bayes' classifier. Sequences of symbols are grouped together in the first step. To this end, we create n-grams for the normal alteration of the stream. The sequence of symbols from the discussion side, $s_1, s_2, \dots \dots s_{l+1}$, is changed into a sequence of bigrams of symbols

$x_1 = s_1 - s_2,\ x_2 = s_2 - s_3, \dots \dots \dots\ x_l = s_l - s_{l+1}$.The probabilities of the n-grams can be computed by assuming bi-grams of symbols, where the special bi-grams are assigned $d_1, \dots \dots, d_m$:

$$p\left(\frac{d_j}{X}\right) = \frac{\#\left(x_i = \frac{d_j}{X}\right)}{\sum_k \#\left(x_i = \frac{d_k}{X}\right)} \quad (3)$$

where $\#\left(x_i = \frac{d_j}{x}\right)$ indicates the number of symbols in Sequence X identical to d_j.

The second step of kernel construction is based on the selection of the document components containing term weightings for the entries of vector $v = [p(d_1)\ \dots\ p(d_m)\ p(d_1 - d_1)\ \dots\ p(d_m - d_m)]^T$ and the normalization of the resulting vector. By term weighting, we mean that for each entry v_i, of vector v, we multiply it by a collection (or background) component, w_i for that entry [17,18].

Speaker specific Log-Likelihood Ratio (LLR) Weighting

Let us consider two speaker-specific items of information for classification. We further assume that the succession (for fixed n) of each speech can be analysed by $X = x_1, x_2, \dots \dots, x_l$ and $Y = y_1, y_2, \dots \dots, y_m$, respectively [19, 20]. We also expect an expansive quantity of symbols β to correspond to the number of individuals. This can be represented as the positions of n-grams in the foundation as $d_1, \dots \dots, d_m$ [21]. It can be assembled as a regular form for every speaker consisting of the likelihood of n-grams, $p\left(d_i/_X\right)$ and $p\left(d_i/_Y\right)$. Following this, we calculate the proportional probability as in normal check frameworks. A linear approach to proportional probability calculation is in the form of kernels:

$$\frac{p(X/_Y)}{p(X/_B)} = \frac{p(x_1, \dots \dots . x_l/_Y)}{p(x_1, \dots \dots . x_l/_B)} = \prod_{i=l}^{l} \frac{p(x_i/_Y)}{p(x_i/_B)} \quad (4)$$

Considering the log of proportional probability standardized by the quantity of perceptions,

$$\begin{aligned} \text{score} &= \frac{1}{l}\sum_{i=1}^{l} \log\left(\frac{p\left(\frac{x_i}{Y}\right)}{p\left(\frac{x_i}{B}\right)}\right) \\ &= \sum_{j=1}^{M} \frac{\#\left(x_i = \frac{d_j}{X}\right)}{l} \log\left(\frac{p\left(\frac{d_j}{Y}\right)}{p\left(\frac{d_j}{B}\right)}\right) \\ &= \sum_{j=1}^{M} p\left(\frac{d_j}{X}\right) \log\left(\frac{p\left(\frac{d_j}{Y}\right)}{p\left(\frac{d_j}{B}\right)}\right) \end{aligned} \quad (5)$$

The above can be modified by inverting the parts of the two successions and averaging the scores. Numerous solutions are available to obtain a kernel from the pseudo-kernel in (5). One probability involves framing a direct guess of the capacity of the pseudo-kernel. Any kernel instigates a separation with the accompanying character by the following identity:

$$D[x,y]^2 = [K\{x,x\} + K\{y,y\} - 2K\{x,y\}] \quad (6)$$

where $D(x,y)$ = The maximum margin criterion of SVM hyperplane.

In the event that we obtain an estimation of $D(x,y)$ that near the edge, we can effectively pick bolster vectors and fitting estimations of α_i in (1), i.e. the principle part involves a strategy to pick supporting vectors and determine the most extreme edge such that a precise separation metric is not required.

The Taylor series expansion of (5) to linearize the log

functions by $\log(x) \approx x - 1$ is given by:

$$\text{score} \approx \sum_{j=1}^{M} p\left(\frac{d_j}{X}\right) \frac{p\left(d_j/Y\right)}{p\left(d_j/B\right)} - \sum_{j=1}^{M} p\left(\frac{d_j}{X}\right)$$

$$= \sum_{j=1}^{M} p\left(\frac{d_j}{X}\right) \frac{p\left(\frac{d_j}{Y}\right)}{p\left(\frac{d_j}{B}\right)} - 1$$

$$= \sum_{j=1}^{M} \frac{p\left(\frac{d_j}{X}\right)}{\sqrt{p\left(\frac{d_j}{B}\right)}} \frac{p\left(\frac{d_j}{Y}\right)}{\sqrt{p\left(\frac{d_j}{B}\right)}} - 1 \quad (7)$$

Practically equivalent to the data recovery writing in [22], we can express the outcome in (7) in vector structure. To start with, we develop a vector portraying the discussion side

$$V_X = \left[p\left(\frac{d_1}{X}\right) \dots \dots p\left(\frac{d_M}{X}\right)\right]^t \quad (8)$$

The sections of V are then weighted with a slanting framework D defined as $D_{j,j} = 1 \Big/ \sqrt{p\left(\frac{d_j}{B}\right)}$. The final kernel can be written as;

$$K(X, Y) = \left(D_{V_X}\right)^t \left(D_{V_Y}\right) \quad (9)$$

Similar to data recovery writing, we call (9) a term-frequency kernel LLR (TFLLR).

Term Frequency and Inverse Document Frequency (TFIDF) Weighting

The term frequency $TF(d_i)$ is the number of occurrences of a given word in d_i, an archive. The accumulation segment is some measure of the shared characteristic of the word in every one of the archives in the accumulation. A substitute strategy for term weighting might be inferred by utilizing the accompanying technique as follows: Let us consider two speaker utterances s_1 and s_2, where the n-grams for the utterances are t_1 to t_n and u_1 to u_m, respectively [23]. A unique model based on speaker utterances can be built consisting of n-gram $p(d_i/s_j)$, where the unique set of n-grams is denoted by d_1 to d_m. The likelihood ratio of the utterances is then computed. Let $DF(d_i)$ be the number of sides in conversation where a particular n-gram d_i, is observed; then, a TFIDF can be mathematically represented as [24, 25]:

$$TF(d_i) \log\left(\frac{\#\text{of conversation in background}}{DF(d_i)}\right) \quad (10)$$

The kernel in (7) is fundamentally the same as strategies used in data recovery writing with some minor contrasts. We initially use probabilities instead of vector sections in (8). This is an alternative standardization (1-norm) to the Euclidean standard. The weighting term in (7) is based on the likelihood of an n-gram background

Standard IDF simply measures whether a word occurs in a specific report. Our plan in referring to data recovery is twofold. First, we need to clarify that our linearization of the proportional probability prompts a comparable methodology utilized as a part of data recovery. Second, we can obtain insights from data recovery to enhance our methodology.

Generalizations and their role

In (7), the kernel is summed up in a straightforward manner. There are a few motivations for this speculation. In (7), the foundation weighting term can turn out to be entirely expansive if a given n-gram occurs rarely in a record. In this case, the likelihood evaluation of $p\left(\frac{d_j}{B}\right)$ might be inaccurate. We might want to confine the impact of any one section in the given vector V in (9). We then revise the slanting terms in (9) as:

$$D_{j,j} = \sqrt{\frac{1}{p\left(d_j/B\right)}} \quad (11)$$

The mapping is in the range 1 to ∞. Joining the past two motivations mentioned above, it is common to utilize the corner-to-corner weight of

$$D_{j,j} = \min\left(C_j, g_j\left(\frac{1}{p\left(d_j/B\right)}\right)\right) \quad (12)$$

where $g_j(.)$ is a function that squashes the dynamic range. For $g_j(x) = \sqrt{x}$, we get our TFLLR part. Moreover, as in TFIDF, we can utilize $g_j(x) = \log(x) + 1$. This variable weighting might be useful in case we consider the certainty of our calculations of the probabilities of the out-of-sight information set of n-grams.

IMPLEMENTATION

A few usage traps can improve the utilization of an SVM classifier in preparation and scoring of data. The SVM-based model of ASR can be improved by consolidating the support vectors in (1). If we accept we have the SVM developments of $V_i$in (9),

$$f(V) = \sum_{i=1}^{N} \alpha_i y_i V_i^t V + b$$

$$= V^t\left(\sum_{i=1}^{N}\alpha_i y_i V_i\right) + b \quad = V^t W + b \qquad (13)$$

where $W = \left(\left(\sum_{i=1}^{N}\alpha_i y_i V_i\right)\right)$.

EXPERIMENTAL SETUP

Experiments for speaker-specific phone sequences and SVM telephone-based speaker recognition experiments were performed on the NIST 2005 SRE extended data task [26]. The corpus used was a combination of phases II and III of the Switchboard-2 corpora [27].

Each potential training utterance in the NIST extended data task consisted of a conversation side that was nominally 5 minutes long, recorded over a landline telephone. Each conversation side consisted of a speaker having a conversation on a topic selected by an automatic operator; conversations were typically between people who did not know each other.

For training and testing, the jacknife approach was used to increase the number of tests. The data was divided into 10 splits. For training, a given split contained speakers to be recognised (target speakers) and impostor speakers; the remaining splits could be used to construct models describing the statistics of the general population—a "background" model. For example, when conducting tests on split 1, splits 2-10 could be used to construct a background.

A speaker model was trained by using statistics from 1, 2, 4, 8 and 16 conversation sides. This simulated a situation where the system could use longer-term statistics and become "familiar" with the individual; this longer-term training allowed one to explore techniques which might mimic more what human listeners perceive about an individual's speech habits. A large number of speakers and tests were available; for instance, for eight conversation trainings, 739 distinct target speakers were used, and 11, 171 true trials and 17, 736 false trials were performed. For additional information on the training/testing structure, we refer to the NIST extended data task description [28].

The objective of our speaker recognition experiments objective was to exhibit the effectiveness of applying an SVM to HLSSF; we did not attempt to deliver high-level feature frameworks.

SVM Phone System

We considered English (EN), Mandarin (MA) and German (GE). For each language, n-gram probabilities were calculated using (3). Then, the probabilities were vectorized as in (8) using a sparse representation, i.e. we stored the indices and the probabilities of nonzero entries only.

We then used TFLLR probability. A score was finally calculated for each language using the linear kernel, and the weighted linear combination technique was used to fuse the scores.

Speaker Specific Phone Sequence Extraction

Telephone speech signal-based speaker-specific feature extraction for a caller-check procedure was carried out using the telephone based ASR framework from the PPRLM language-recognisable proof framework. PPRLM utilizes a Mel frequency cepstral coefficient (MFCC) [29, 30, 31,32] front end with attached delta coefficients. Every telephone is modelled in a gender-independent, language-independent and text-independent way using a three-state hidden Markov model (HMM) [8]. This dataset contained four Gaussians: the left and the right ones constituted one class, whereas the other two constituted the other class. It was obvious that a global linear SVM could not correctly separate the data from the two classes. The linear kernel SVM was able to classify the data correctly by learning a non-linear decision boundary [33, 34]. Our proposed SVM partitioned the data into two clusters (the dark black curve is the boundary between the two clusters), and trained a linear SVM in each cluster. We see that both SVM could separate the linearly non-separable data. Figure 1 shows classifiers learned based on the linear SVM.

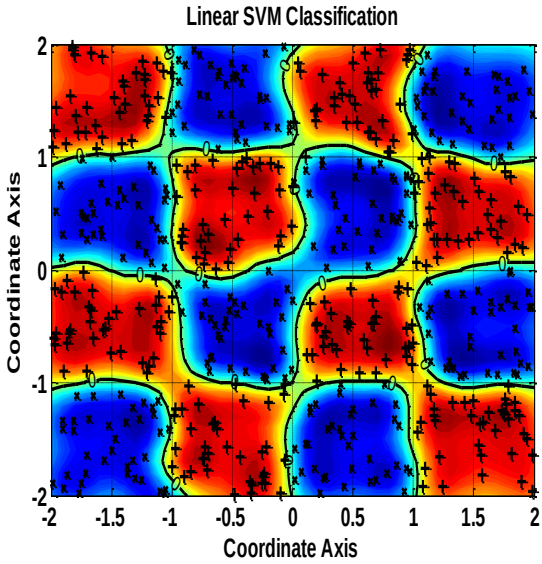

Figure 1: Classifiers learned on the linear SVM

Phone Experiments Corpus

The SVM framework was constructed using the NIST 2005 SRE corpus. Nine sets of corpora were considered, one for training and eight for testing the ASR system. To test the reliability of the ASR system, we used 1,672 true and 14,406 false speakers. All experiments were conducted on Fisher's corpus. We also considered some non-English languages, such as Hindi, Tamil and Telugu. The consequent foundation had 4,171 discussion sides.

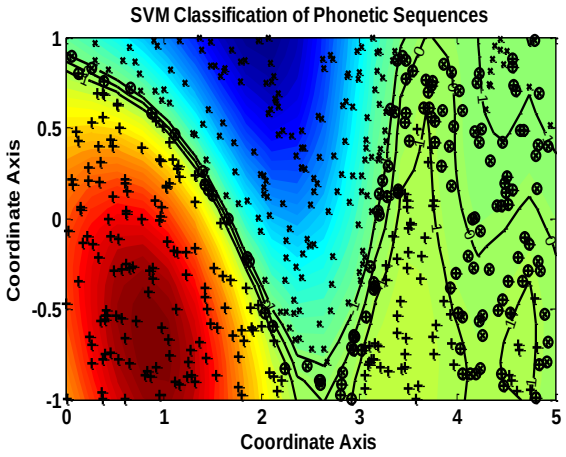

Figure 2: Classifiers learned on the linear SVM with phone experiments corpus

Every discussion side in the NIST SRE dataset lasted four seconds. Each discussion side consisted of a speaker having a discussion on a subject chosen by a programmed administrator; discussions were ordinarily between people who did not know each other. Figure 2 shows the classifiers learned on the phone experiments' corpus.

SVM Phone Training

The SVM-based ASR system was trained per the task definition of the NIST SRE database. A corpus of the NIST SRE database was utilized for testing and training the SVM ASR model. One corpus was used for training and remaining eight for testing to fulfil the requirements of the one verses all strategy [35]. The SVM-based ASR categorised the target speakers as "1". The speakers who did not belong to the registered database were categorised as "-1". This technique guaranteed that individuals considered false speakers were eliminated from the train/test database. Figure 3 shows classifiers learned on the linear kernel-based SVM phone training target estimation of either "1" or "-1".

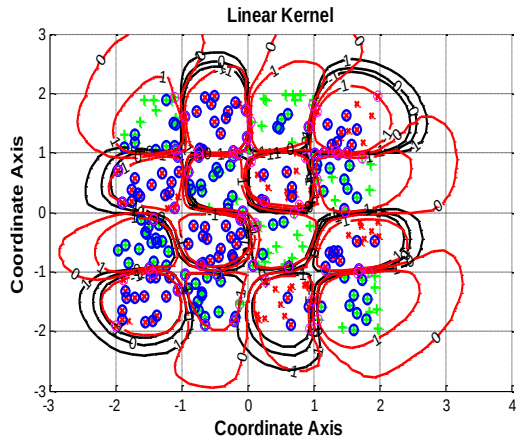

Figure 3: Learned classifiers on the linear kernel based SVM phone training

SVM Based ASR and Its outcomes of Phone Experiments

An experiment based on SVM and the scoring framework of ASR was conducted as well. The normalized weights of SVM scores from different language symbolisations were utilised in ASR. In Fig. 4, we compare the DET curves of different values of the SVM ASR on the NIST SRE 2005 corpus.

We compared the results of the ASR systems on the NIST SRE 2005 corpus for SVM Bigram 1, SVM Bigram Lattice and SVM Trigram lattice. The EER of SVM Bigram 1 was 12%, that of SVM Bigram Lattice was 9% and that of SVM Trigram Lattice was 6%. The performance of SVM Trigram Lattice was the best considering standard SVM configuration. The lower computational complexity resulted in excellent performance of the linear SVM Trigram Lattice-based ASR system.

We compared the results of ASR systems on the NIST SRE 2005 corpus using SVM-based TFLLR. The EER for German was 15%, that for Mandarin was 8% and for English was 6%. The performance of the linear TFLLR on the English corpus was the best.

Table 1: EER comparison of 1-best and lattice-based SVM models

Model	Decoding	n-gram Order	Language	EER %
SVM	1-Best	2	GE	10.69
SVM	Lattice	2	MA	9.65
SVM	Lattice	3	EG	8.42

Table 1 shows the comparison of EERs of the SVM Lattice-based model. The SVM-based ASR of n-gram order 3 was superior to others.

EXPERIMENTAL RESULTS

In Figure 4. we compare the results of ASR systems on the NIST SRE 2005 corpus for TFLLR $C_j =$ Infinity , TFLLR $C_j = 5$ and TFLLR $C_j = 1$. The EER of TFLLR $C_j =$ Infinity was 9.5%, that of TFLLR $C_j = 5$ was 4.5% and TFLLR $C_j = 1$ was 2.5%. TFLLR $C_j = 1$ yielded the best performance in the standard configuration. Its lower computational complexity led to the excellent performance of the linear GMM super vector kernel-based ASR system.

Table 2: EER comparison of different weighting methods of NIST SRE 2005

Method and Selected C_J	EER %
TFLLR and $C_J = \infty$	9.5
TFLLR and $C_J = 10$	6.5
TFLLR and $C_J = 5$	4.5
Linear and $C_J = 1$	2.5

Table 2 compares the performance of ASR systems with different weighting methods on the NIST SRE 2005 corpus. The performance of ASR with TFLLR $C_j = 1$ was the best in standard configuration.

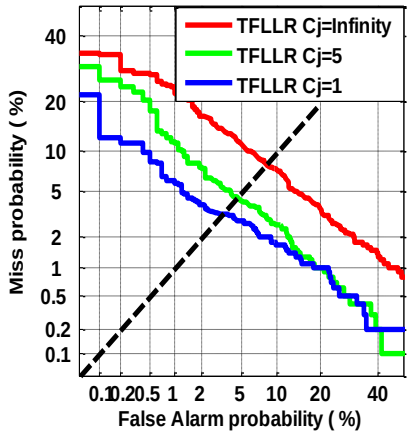

Figure 4: Examination of various weighting sorts for a SVM Word framework

CONCLUSION

Extensive research work has been done in recent years on ASR with limited speech data. This paper introduced an algorithm for speaker-specific feature extraction in an HLSSF-based ASR system. The NIST SRE 2005 evaluation corpus was used to test the proposed algorithm. The results showed that ASR efficiency improved with the application of HLSSF. The efficiency of the SVM and HLSSF-based ASR was compared in two cases, and we found that the Linear and $C_J = 1$ improves significantly. We found that the efficiency of ASR with $C_J = 1$ improved compared to that with $C_J = 10$ and $C_J = 5$.

Similar efficiency trends have been found using different modelling techniques. The EER performances of ASR with TFLLR and C_J =10 on the NIST SRE 2005 corpus was 9.5%, TFLLR with C_J =5 was 4.5% and TFLLR with C_J =1 was 2.5%. We observed a 2% EER improvement in the case of TFLLR and C_J =5, and 4% improvement in the case of TFLLR and C_J =1. As part of future work to enhance ASR efficiency, we plan to alter the algorithm in terms of mid-frequency components.

REFERENCES

[1] E. Formisano, F. De Martino, M. Bonte, and R. Goebel. "Who is saying what Brainbased decoding of human voice and speech". Science, vol. 322 , pp. 970-973, 2008.

[2] A. O. Hatch, S. S. Kajarekar, and A. Stolcke. "Within-class covariance normalization for SVM-based speaker recognition". in Proc. Interspeech, Pittsburgh, PA, pp. 1471-1474, 2006.

[3] G. Doddington. "Speaker recognition based on idiolectal differences between speakers". in Proc. Eurospeech, pp. 2521-2524, 2001.

[4] N. Dehak, P. Kenny, R. Dehak, O. Glembek, P. Dumouchel, L. Burget, V. Hubeika, and F. Castaldo. "Support vector machines and joint factor analysis for speaker verification". in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP'09), pp. 4237-4240, 2009.

[5] G. Doddington. "Speaker recognition based on idiolectal differences between speakers". in Proceedings of Eurospeech, pp. 2521-2524, 2001.

[6] Walter D. Andrews, Mary A. Kohler, Joseph P. Campbell, John J. Godfrey, and Jaime Hern´andez-Cordero. "Gender-dependent phonetic refraction for speaker recognition". in Proceedings of the International Conference on Acoustics Speech and Signal Processing, pp. I149-I153, 2002.

[7] Gauvain, J., Messaoudi, A., Schwenk, H.. "Language recognition using phone lattices". in: Proc. ICSLP, 2004.

[8] Osman Büyük.: "Sentence-HMM state-based i-vector/PLDA modelling for improved performance in text dependent single utterance speaker verification" IET Signal Processing, Vol 10, Issue 8 , pp. 918 - 923, 2016.

[9] Torres-Carrasquillo, P.A., Reynolds, D.A., Deller, J.R,. "Language identification using gaussian mixture model tokenization". In: Proc. ICASSP, IEEE, 2002.

[10] Siniscalchi, S.M., Reed, J., Svendsen, T., Lee, C.H.: "Exploiting context-dependency and acoustic resolution of universal speech attribute models in spoken language recognition". In: Proc. INTERSPEECH, pp. 2718-2721, 2010.

[11] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren,. "A novel scheme for speaker recognition using a phonetically-aware deep neural network". In Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP'14), pp. 1695-1699, 2014.

[12] Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups". IEEE Signal Processing Magazine 29, pp.82-97, 2012.

[13] L. Nguyen, S. Matsoukas, J. Davenport, F. Kubala, R. Schwartz, and J. Makhoul,. "Progress in transcription of broadcast news using byblos". Speech Commun., vol. 38, pp. 213-230, 2002.

[14] C. M. Bishop and N. M. Nasrabadi. "Pattern Recognition and Machine Learning", New York: Springer, 2006.

[15] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. "Front-end factor analysis for speaker verification". IEEE Trans. Audio, Speech, Lang. Process., vol. 19, no. 4, pp. 788-798, 2011.

[16] Sandro Cumani, Pietro Laface. "Analysis of Large-Scale SVM Training Algorithms for Language and Speaker Recognition". IEEE Transactions on Audio, Speech, and Language Processing, Vol. 20, Issue: 5, pp. 1585-1596, 2012.

[17] John H.L. Hansen and Taufiq Hasan. "Speaker Recognition by Machines and Humans a tutorial review". IEEE signal processing magazine November, pp.74-96, 2015.

[18] C. Cortes and V. Vapnik. "Support-vector networks". Mach. Learning, vol. 20, no. 3, pp. 273-297, , 1995.

[19] W. M. Campbell. "Generalized linear discriminant sequence kernels for speaker recognition". In Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP'02), pp. 161-164.

[20] G. S. Morrison. "Distinguishing between science and pseudoscience in forensic acoustics". In Proc. Meetings on Acoustics, Vol. 19, 2013.

[21] G. S. Morrison. "Forensic voice comparison". in Expert Evidence 99, 1 ed. London: Thompson Reuters, Chap. 99, pp. 1051-1058, 2010.

[22] T. Joachims. "Learning to Classify Text Using Support Vector Machines". Norwell, MA: Kluwer, 2002.

[23] P. Rose. "Forensic Speaker Identification". Boca Raton, FL: CRC Press, 2004.

[24] Hsiang-Fu Yu, Fang-Lan Huang, and Chih-Jen Lin,. "Dual coordinate descent methods for logistic regression and maximum entropy models". Machine Learning, 85(1-2), pp 41-75, 2011.

[25] Charles S. Ahn. "Automatically detecting authors native language". Ph.D. thesis, Monterey, California. Naval Postgraduate School.

[26] Joel Tetreault, Daniel Blanchard, and Aoife Cahill. "Summary report on the first shared task on native language identification". In Proceedings of the Eighth Workshop on Building Educational Applications Using NLP, Atlanta, GA, USA, June. Association for Computational Linguistics, 2013.

[27] M. Przybocki and A. Martin. "The NIST year 2003 speaker recognition evaluation plan". http://www.nist.gov/speech/tests/spk/2003/index.htm, 2003.

[28] Linguistic Data Consortium, "Switchboard-2 corpora," http://www.ldc.upenn.edu.

[29] S. Singh, Abhay Kumar, David Raju Kolluri. "Efficient Modelling Technique based Speaker Recognition under Limited Speech Data". International Journal of Image, Graphics and Signal Processing(IJIGSP), ISSN: 2074-9074, Vol.8, No.11, pp.41-48, 2016.

[30] S.Singh, Mansour. H. Assaf and Abhay Kumar. "A Novel Algorithm of Sparse Representations for Speech Compression/Enhancement and Its Application in Speaker Recognition System". International Journal of Computational and Applied Mathematics. ISSN 1819-4966, Volume 11, pp. 89-104, 2016.

[31] S.Singh and Dr. E.G. Rajan. "Application Of Different Filters In Mel Frequency Cepstral Coefficients Feature Extraction And Fuzzy Vector Quantization Approach In Speaker Recognition". International Journal of Engineering Research & Technology, Vol. 2 Issue 6, June - 2013

[32] S.Singh and Ajeet Singh. "Accuracy Comparison using Different Modeling Techniques under Limited Speech Data of Speaker Recognition Systems". Global Journal of Science Frontier Research: F Mathematics and Decision Sciences, ISSN: 0975-5896, Volume 16, Issue 2, Version 1.0, pp.1-17, 2016.

[33] Yi-Hsiang Chao. "Speaker identification using pairwise log-likelihood ratio measures". Fuzzy Systems and Knowledge Discovery (FSKD), 9th International Conference on 29-31, pp. 1248-1251, 2012.

[34] C. M. Bishop and N. M. Nasrabadi. "Pattern Recognition and Machine Learning". New York: Springer, 2006.

[35] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. "Front-end factor analysis for speaker verification". IEEE Trans. Audio, Speech, Lang. Process., vol. 19, no. 4, pp. 788-798, 2011.