A FAST PHONEME RECOGNITION SYSTEM BASED ON SPARSE REPRESENTATION OF TEST UTTERANCES

Armin Saeb[1], *Farbod Razzazi*[2], *Massoud Babaei-Zadeh*[3]

[1]Electrical Engineering Department, Islamic Azad University, Shahr-e-Rey Branch, Tehran, Iran.
[2]Electrical and Computer Engineering Department, Islamic Azad University, Science and Research Branch, Tehran, Iran.
[3]Electrical Engineering Department, Sharif University of Technology, Tehran, Iran.

ABSTRACT

In this paper, a fast phoneme recognition system is introduced based on sparse representation. In this approach, the phoneme recognition is fulfilled by Viterbi decoding on support vector machines (SVM) output probability estimates. The candidate classes for classification are adaptively pruned by a k-dimensional (KD) tree search followed by a sparse representation (SR) based class selector with adaptive number of classes. We applied the proposed approach to introduce a phoneme recognition system and compared it with some well-known phoneme recognition systems according to accuracy and complexity issues. By this approach, we obtain competitive phoneme error rate with promising computational complexity in comparison with the state of the art phoneme recognition systems which causes this approach become a suitable candidate for automatic speech recognition (ASR) applications.

Index Terms— KD-Tree, Phoneme Recognition, Sparse Representation, Support Vector Machines, Viterbi Search, Sparse Class Selector

1. INTRODUCTION

Phoneme recognition, as the procedure of estimating a sequence of phonetic labels, given a sequence of acoustic feature vectors, is the core block of most of ASR systems. There are many approaches for phoneme recognition like hidden Markov Model (HMM) based approaches [1] and online learning algorithms [2]-[4], which are trying to decrease the phoneme error rate. There are two issues that should be reconsidered in the previous proposed approaches. First, the computational complexity of the systems has drastically been grown in state of the art low error rate recognizers. Second, most of the proposed models are generally trained by a predefined training set of utterances, which makes the system not reliable in speaker and environmental conditions of the test utterance.

Sparse Representation (SR) is a technique to represent a signal by a small number of basic signals (atoms) [5]. In recent years, SR has been successfully used for some signal processing applications [6]-[8] including phoneme classification and recognition [9]-[11]. In SR, an $n \times 1$ vector $\mathbf{y}$ is approximated by a linear combination of a few vectors which are selected from m basic vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ where $n < m$, to have $\mathbf{y} = \mathbf{X}.\boldsymbol{\lambda}$ where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]$. The coefficients vector $\boldsymbol{\lambda}$ should be obtained by minimizing its l^0 norm or some other approximating criteria.

SR seems to be a promising approach for speech processing problems due to the existence of sufficient training utterances which are phonetically labeled well. Therefore, the chance of similarity between the test vector and a sparse set of training vectors increases and the test vector can be represented by a linear combination of this sparse set. Sainath *et al.* have used this approach for phoneme classification [9] and have extended their method to phoneme recognition for large vocabulary continuous speech recognition (LVCSR) [10]. Gemmeke *et al.* have employed SR as a missing data technique (MDT) to estimate clean speech features from noisy speech signal [6]. Li *et al.* have applied SR to estimate the clean feature vector from noisy feature vector and have used it for in-car speech recognition [7]. In [11], we have introduced a fast phoneme classifier based on smoothed l^0 norm (SL0) sparse representation [12]. Our approach has been based on training a classifier based on a very limited selection set of training samples. Then, we have used SR approach as a class selector (not as a classifier) and have selected two best classes (more probable classes). Finally, we have applied the training vectors of these two best classes for training SVM classifier and have forced this classifier to predict the test vector's label among these two classes. We have applied this approach for phoneme classification [11] and we have obtained promising results. Besides, in [13] we have

compared the computational complexity of this algorithm with well-known classifiers and we have shown that it is considerably fast. Although our proposed algorithm seems to be a suitable candidate for classification applications, it is not a proper approach for time series recognition (e. g. phoneme recognition). In addition, our previous algorithm imposes to select two best classes and forces the SVM classifier to select the correct class between them. Therefore, some correct classes are omitted from the classification procedure, increasing the overall error rate.

In this paper, we propose an adaptive and fast continuous phoneme recognition system based on our previously proposed phoneme classifier [11]. Firstly, for each frame of the test utterance, an N-best list of classes is prepared by using a tree search strategy followed by an SL0 sparse representation class selector where the parameter N is selected adaptively. Subsequently, an N-class SVM classifier is trained by the selected vectors. Then, the trained classifier is employed to estimate the likelihood probabilities of the test utterance in the N-best candidate classes. Finally, the estimated probabilities are used to generate a matrix which each column includes probabilities of all classes given a frame. This matrix is used as a trellis diagram which is applied to Viterbi decoding algorithm to predict the best phoneme sequence per each test speech utterance. Simulations show that this method results in a noticeable low error rate in both frame and phoneme basis with a fair computational complexity.

The rest of the paper is organized as follows. In section 2, we explain the proposed approach. The experimental results of evaluation of the idea on a phoneme recognition benchmark are presented in section 3 in addition to comparison with other well known phoneme recognition systems. Finally, section 4 concludes the paper.

2. PROPOSED APPROACH

Fig. 1 shows the block diagram of the proposed approach. Firstly, we chunked the training and testing utterances into frames. Each frame was represented by standard MFCC features to form training and testing vectors. In the recognition phase, a set of m training neighbors of each test vector was obtained using KD-tree search strategy. We constructed the search tree using all training vectors in a batch offline procedure. These m neighbors formed the matrix X_t which was used in a standard SR problem:

$$\boldsymbol{\lambda}^{*} = Arg\,Min\|\boldsymbol{\lambda}\|_0 \quad st. \quad \mathbf{y}_t = \mathbf{X}_t.\boldsymbol{\lambda} \tag{1}$$

To solve (1), we used SL0 algorithm which is a sparse decomposition approach without relaxing l^0 norm with l^1 norm. Instead, this algorithm substitutes $\|\boldsymbol{\lambda}\|_0$ in (1) with a suitable approximating continuous function of $\boldsymbol{\lambda}$. Therefore, the following problem will be we solved instead of (1) [12]:

$$\boldsymbol{\lambda}^{*} = ArgMin[m - F_{\sigma}(\boldsymbol{\lambda})] \qquad s.t. \quad \mathbf{y}_t = \mathbf{X}_t.\boldsymbol{\lambda} \tag{2}$$

where $F_{\sigma}(\boldsymbol{\lambda})$ is a smooth differentiable function of $\boldsymbol{\lambda}$ as an approximation for $m - \|\boldsymbol{\lambda}\|_0$, in which σ is the smoothness controlling parameter, affecting the accuracy of the approximation [12].

We used the coefficient vector $\boldsymbol{\lambda}$ to rank the labels of classes to determine the best classes list $\boldsymbol{\beta}$. If we define the labels of training vectors of matrix $\mathbf{X}_t$ as $d_k \;\; k = 1,2,...,m$ and define c_i as one of the K possible labels of classes, the class selector criterion is defined as:

$$\eta_i = \sum_{k=1}^{m} \lambda_k^2 \delta[d_k - c_i] \quad i = 1,2,...K \tag{3}$$

where $\delta[.]$ is the unit impulse function.

We sorted η_i s decreasingly and named them η'_i s. Therefore:

$$\eta'_1 > \eta'_2 > ... > \eta'_K \tag{4}$$

We used η'_i to determine the best classes and formed the ranked class vector $\boldsymbol{\beta} = [c'_1, c'_2, ..., c'_K]$, where c'_i is the class label corresponding to η'_i. Then, to form the N-best classes list $\boldsymbol{\beta}_N = [c'_1, c'_2, ..., c'_N]$, we used:

$$N = Arg\,Min(k) \quad st. \quad (\eta'_1 / \eta'_k) > 2^{TH} \tag{5}$$

where TH is a threshold which controls the average number of classes. If η'_1 is much greater than other elements of $\boldsymbol{\beta}$, it means that SL0 decision is appropriate and reliable. Therefore, $N = 2$ would be selected and binary SVM classifier will be used. Otherwise, if η'_1 is not significantly greater than other elements of $\boldsymbol{\beta}$, it means that SL0 outputs is not confident and therefore, we selected more classes for SVM classification.

We used the adaptive parameters N and $\boldsymbol{\beta}_N$ to select the corresponding training vectors from a predefined number of closest neighbors of the test vector to construct the matrix $\mathbf{Z}_t$ which is used to train the N class SVM classifier (or any arbitrary classifier). The SVM classifier's outputs are the probability estimates which are calculated based on the approach that has been presented by Wu *et al.* [14].

We integrated and smoothed the estimated probability arrays of all the frames of a test utterance to construct the trellis diagram matrix $\mathbf{P}$. For smoothing, we added a bias to all elements of $\mathbf{P}$ to avoid zero probability of some states. We applied this matrix as the observation probabilities of Viterbi

decoding algorithms. In addition, as the states of the constructed trellis corresponds to phonemes, we calculated the state transition probability matrix $\mathbf{A}$ and initial state distribution array $\boldsymbol{\pi}$ from the training data transcription in offline and applied them to the decoding algorithm. The phoneme sequence is the output of the Viterbi decoder.

3. EXPERIMENTS

To assess the proposed phoneme recognition system, a set of experiments were conducted on extracted features from TIMIT database [15]. In this paper, we used 3696 utterances of 462 speakers from the training set (all training set excluding SA1 and SA2 utterances). The evaluation set was standard core test set of 50 speakers and 192 utterances. The test was employed in accordance with standard examinations on TIMIT [16]. Firstly, the 61 phonetic labels were converted to 48 labels. Then, the acoustic model was trained with the 48 labels set. Finally, the 48 labels set was collapsed into a smaller set of 39 labels to correctly evaluate the recognition performance. The extracted features were the standard 12 MFCC features and log energy, with their first and second derivatives, to have a 39 dimensional vector per each frame. The frame size is 25ms, with a frame-shift of 10ms. By this approach, we extracted 1120114 and 57213 training and testing vectors respectively. We conducted the experiments based on the architecture of Fig. 1. The number of neighbors in KD-Tree search (m) was set to 200 [9].

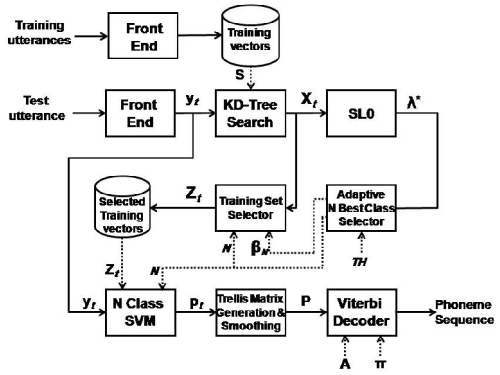

Fig. 1. Block diagram of the proposed phoneme recognizer.

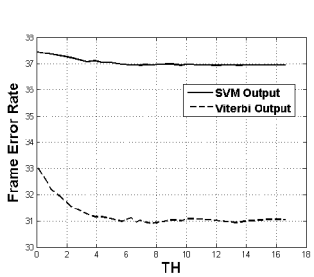

Fig. 2. Frame error rate of SVM classifier and Viterbi search algorithm

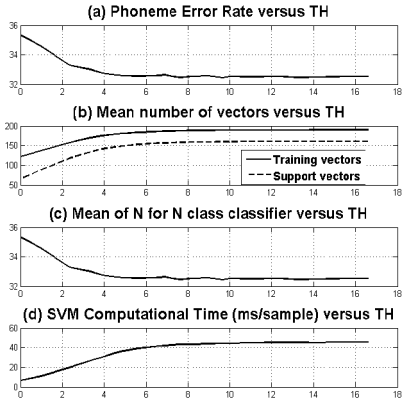

Fig. 3. Proposed algorithm's specifications versus TH

Experiment 1. In the first experiment, we evaluated the frame error rate (FER) of the proposed approach at the SVM and Viterbi outputs versus TH changes. As shown in Fig. 2, the FER decreases as TH increases. This is due to the fact that the increase of TH results in the increase in the number of selected N-best classes which makes the phoneme recognition system more accurate. In addition, it can be observed that FER is saturated while TH rises to more than 5. Therefore, $TH = 5$ would be a suitable value for our approach. In addition, approximately 6% reduction in FER can be observed by using Viterbi decoder.

Experiment 2. In the second experiment, we assessed other evaluating parameters of the proposed classifier versus TH changes. As shown in Fig. 3, the phoneme error rate (PER) decreases as TH increases. However, this makes the algorithm slower. The effect of TH on other parts of the algorithm was not investigated; because they are independent of TH. The saturation effect can also be observed on the number of training vectors, the number of support vectors and the number of selected classes as TH goes more than 5.

Experiment 3. In the third experiment, we reported our proposed algorithm's edition errors on the empirically optimized $TH = 5$. As shown in Table I, the main contribution of the error belongs to substitutions which form nearly %60 of errors.

Experiment 4. In the fourth experiment, we compared the computational time of the proposed phoneme recognition system with some well-known algorithms including SVM following Viterbi decoder (without KD-search and SL0), SR-SL0 (proposed approach excluding SVM). We used a PC with Intel core i5 2.53GHz CPU and 4GB RAM. We used LIBSVM package for SVM [17], HMM toolbox for Viterbi decoder [18] and SL0 package [19] for SL0. We used Radial Basis Function (RBF) kernel for SVM and optimized RBF parameters to minimize the error rate except

for SVM2 which we intended to decrease the computational complexity; therefore, we selected a wide RBF to have a small number of support vectors. As indicated in Table 2, our proposed approach results a phoneme recognition system which is fast and with promising FER and PER in comparison with some well-known algorithms. We should mention that although the first algorithm in Table 2 has a better PER comparing to our proposed algorithm, it suffers from very high complexity and is not a good candidate for phoneme recognition systems. In the SVM1, almost %80 of training samples (89600) are selected as the support vectors. The training period of this classifier is approximately two weeks in our PC and its recognition time is almost 2.5 times of our proposed approach. As the SVM becomes simpler (e. g. SVM2 with 56000 support vectors) PER increases and becomes more than our proposed algorithm; even with more computational complexity. The third row in Table 2 reports a recognizer that excludes SVM from the proposed approach to have an SR classifier following Viterbi decoder. Although this algorithm is a little faster than our proposed phoneme recognition system, but its FER and PER is significantly more than ours.

Experiment 5. In the fifth experiment, we analyzed the computational time of the proposed phoneme recognition system. As shown in Table 3, most of the time was spent to search of the best neighbors of the test sample. Therefore, it can be replaced by other search algorithms with less complexity [10] and the proposed phoneme recognition system has a good potential to increase its speed.

Finally, we quoted the results of some other state of the art phoneme recognizers and compared with the proposed system in Table 4. It can be observed that in addition to high speed, our proposed phoneme recognition system has a good FER and PER in comparison with some well-known algorithms which are likely slower than the proposed approach.

4. CONCLUTIONS

In this paper, we introduced a test ensemble adapted fast phoneme recognition system with acceptable phoneme error rate based on sparse representation for adaptive class pruning in the classification phase. The main idea of this approach is the selection of appropriate and variable number of training samples set from the whole training sample set which is adapted to test sample and correspond to the most likely classes. This procedure was implemented by KD-tree search algorithm followed by a fast SR algorithm. We used the results of SR algorithm to identify the number of most likely classes. Then, we trained an adaptive multiclass SVM classifier with these reduced and adapted training samples to extract the probability estimation array. These arrays were used to form a matrix as the trellis diagram per each testing utterance. Finally we applied the Viterbi algorithm to predict the phoneme sequence. Simulation results showed that this approach is fast and accurate enough. In addition, the proposed controllable trade-off between speed and accuracy in the proposed approach is very promising.

Table 1. edition error analysis for the proposed phoneme recognition system and $TH = 5$

Edition Error	Percent
Substitution	56.7
Deletion	32.8
Insertion	10.5

Table 2. recognition time for some phoneme recognition systems

Method	Total recognition time (ms/sample)	%FER	%PER
SVM1+Viterbi	2712.4	28.7	29.6
SVM2+Viterbi	2005.6	33.2	34.4
SR_SL0+Viterbi	1097.2	35.4	37.3
Proposed	**1116.8**	**31**	**32.5**

Table 3. computational time analysis for the proposed phoneme recognition system and $TH = 5$

Algorithm	Computational time (ms/sample)
KD search	1095.1
SL0	1.9
SVM Training	19.385
SVM Prediction	0.25
Viterbi	0.185

Table 4. reported results on TIMIT core test set

Method	%FER	%PER
HMM[1]	39.3	42
LM-HMM[1]	25	30.2
PA[2]	30	33.4
PAC-Bayesian 9frames[3]	26.5	28.6
DIAG 3 frames context[4]	32.3	45.4
APPROX 3 frames context[4]	27.8	27.9
MATCH 3 frames context[4]	28.0	28.0
SVM1	28.7	29.6
SVM2	33.2	34.4
SR-SL0	35.4	37.3
Proposed approach	**31**	**32.5**

5. REFERENCES

[1] C. C. Cheng, F. Sha, and L. K. Saul, "A fast online algorithm for large margin training of continuous-density hidden Markov models," in *Proc. Interspeech,* pp. 668-671, 2009.

[2] K. Crammer, "Efficient online learning with individual learning rates for phoneme sequence recognition," in *Proc. ICASSP,* pp. 4878-4881, 2010.

[3] J. Keshet, D. McAllester, and T. Hazan, "Pac-Baysian approach for minimization of phoneme error rata," in *Proc. ICASSP,* pp. 2224-2227, 2011.

[4] K. Crammer, and D. D. Lee "Online discriminative learning of phoneme recognition via collection of generalized linear models," in *Proc. ICASSP,* pp. 1961-1964, 2012.

[5] M. Elad, "Sparse and redundant representations," Springer Press, 2012.

[6] J. F. Gemmeke, H. V. Hammen, B. Cranen, and L. Boves, "Compressive sensing for missing data imputation in noise robust speech recognition," *IEEE Journal of selected topics in Signal Processing,* vol. 4, no. 2, pp. 272-287, 2010.

[7] W. Li., Y. Zhou, N. Poh, F. Zhou, and Q. Liao, "Feature denoising using joint sparse representation for in-car speech recognition," *IEEE Signal Processing Letters,* vol. 20, Issue 7, pp. 681-684, 2013.

[8] A. Asaei, M. J.Taghizadeh, H. Bourlard, V. Cevher, "Multi-party speech recovery exploiting structured sparsity models," in *Proc. Interspeech,* pp. 185-188, 2011.

[9] T. N. Sainath, A. Carmi, D. Kanevsky, and B. Ramabhadram, "Bayesian compressive sensing for phonetic classification," in *Proc. ICASSP,* pp. 4370-4373, 2009.

[10] T. N. Sainath, B. Ramabhadram, M. Picheny, D. Nahamoo, and D. Kanevsky, "Exemplar-based sparse representation features: from TIMIT to LVCSR," *IEEE Transactions on Audio, Speech and Language Processing,* vol. 19, pp. 2598-2613, 2011.

[11] A. Saeb, and F. Razzazi, "A fast compressive sensing approach for phoneme classification," in *Proc. ICASSP,* pp. 4281-4284, 2012.

[12] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed L0 norm," *IEEE Transaction on Signal Processing,* vol. 57, no. 1, pp. 289-301, 2009.

[13] A. Saeb, F. Razzazi, and M. Babaie-Zadeh, "SR-NBS: A fast sparse representation based N-best class selector for robust phoneme classification," *Engineering Applications of Artificial Intelligence,* vol. 28, pp. 155-164, 2014.

[14] T. F. Wu, C. J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pair-wise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975-1005, 2004.

[15] L. Lemel, R. Kassel, S. Seneff, "Speech database development: design and analysis of the acoustic phonetic corpus," in *Proc. DARPA Workshop on Speech Recognition*, pp. 100-109, 1986.

[16] K. F. Lee, and H. W. Hon, "Speaker independent phone recognition using hidden markov models", *IEEE Transaction on Acoustic, Speech, and Signal Processing*, vol. 37, no. 2, pp. 1641-1648, 1989.

[17] C. C. Chang, and C. J. Lin.(2013, January). LIBSVM:a library for support vector machine. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[18] K. Murphy. (2005, June). Hidden Makov Model (HMM) toolbox. Software available at http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html.

[19] H. Mohimani, M. Babaeie-Zadeh, and C. Jutten. (2010, April). Smoothed L0 (SL0) algorithm for sparse decomposition. Available at http://ee.sharif.edu/~SLzero.