SPARSE CODING FOR SPEECH RECOGNITION

G.S.V.S. Sivaram[1,2]*, Sridhar Krishna Nemala*[1]*, Mounya Elhilali*[1]*, Trac D. Tran*[1]*, Hynek Hermansky*[1,2]

[1]Dept. of Electrical & Computer Engineering,
[2]Human Language Technology, Center of Excellence,
Johns Hopkins University, USA.
e-mail : {*sivaram, nemala, mounya, trac, hynek*}*@jhu.edu*

ABSTRACT

This paper proposes a novel feature extraction technique for speech recognition based on the principles of sparse coding. The idea is to express a spectro-temporal pattern of speech as a linear combination of an overcomplete set of basis functions such that the weights of the linear combination are sparse. These weights (features) are subsequently used for acoustic modeling. We learn a set of overcomplete basis functions (dictionary) from the training set by adopting a previously proposed algorithm which iteratively minimizes the reconstruction error and maximizes the sparsity of weights. Furthermore, features are derived using the learned basis functions by applying the well established principles of compressive sensing. Phoneme recognition experiments show that the proposed features outperform the conventional features in both clean and noisy conditions.

Index Terms— sparse coding, feature extraction, compressive sensing, speech recognition.

1. INTRODUCTION

Multilayer perceptron (MLP) classifier based acoustic modeling has been successfully used in state-of-the-art automatic speech recognition (ASR) systems [1]. It facilitates the correlated features with complex density function to be used as the input acoustic observations. Thus, recent feature extraction techniques have focused on ways to encode the information in the spectro-temporal patterns of speech. Most of the techniques employ simple projection-based approach for encoding information. In other words, features are extracted by simply projecting an input spectro-temporal pattern on a set of two-dimensional patterns which characterize various two-dimensional filters. For instance, a set of two-dimensional Gabor filters are preselected to form multiple feature streams in [2]. Furthermore, two-dimensional filter shapes are learned from the data in a discriminative fashion in [3].

The aforementioned feature extraction techniques bear a close resemblance to the spectro-temporal receptive field (STRF) model for predicting the response of a cortical neuron to the input speech [4]. STRF of a neuron describes the two-dimensional spectro-temporal pattern to which that neuron is most responsive, and the response is obtained by projecting an input pattern on the STRF. However, since it is a linear model, STRFs cannot explain the non-linear behavior exhibited by most cortical neurons. However, it is suggested in [5] that sparse coding could be a potential strategy employed by neurons in the visual cortex to encode images in a non-linear manner. The sparse coding idea has been successfully applied for single channel speaker separation [6].

In this work, we demonstrate the usefulness of sparse coding in deriving features for phoneme recognition. Sparse coding deals with the problem of how to represent a given input spectro-temporal pattern as a linear combination of a minimum number of basis functions in an overcomplete dictionary (i.e., the input dimensionality is typically much less than the number of basis functions or atoms in the dictionary). The weights of the linear combination are used as features for acoustic modeling.

Obtaining features involves two steps:- (i) learning the optimal dictionary of basis functions from the training data and (ii) determining the features from the learned overcomplete set. We train the dictionary in an iterative way using the gradient descent algorithm such that it maximizes the sparsity of the features and minimizes the reconstruction error of the spectro-temporal patterns present in the training data [7]. Once the overcomplete set is found, features corresponding to an input spectro-temporal pattern are obtained by minimizing the l_1 norm of the weights of the linear combination of basis functions subject to the faithful reconstruction of the input spectro-temporal pattern by the linear combination. This l_1 norm minimization technique is well established in the compressive sampling literature and yields a sparse weight (feature) vector [8].

2. LEARNING OVERCOMPLETE SET OF BASIS

The goal of sparse coding is to express a given input pattern as a linear combination of an overcomplete[1] set of basis

[1]We drop the term overcomplete for convenience.

functions such that the weights of the linear combination are sparse. It is trivial to see that the choice of basis functions determines how sparse the weight vector is. Therefore, it is necessary to determine a set of basis functions which capture structure in the data so that any input pattern can be expressed using only few basis functions. The set of basis functions are learned from the training data by solving the following optimization problem, which is adopted from [7], in an iterative fashion.

Suppose that the input pattern $\boldsymbol{s}$ can be approximated as a linear combination of the basis functions $\boldsymbol{\phi_i}$ with weights α_i, then the reconstructed pattern $\hat{\boldsymbol{s}}$ is given by,

$$\hat{\boldsymbol{s}} = \sum_{i=1}^{m} \alpha_i \boldsymbol{\phi_i} \tag{1}$$

The total number of basis functions is indicated by m. Ideally, we want to find the basis which minimizes the expected value of the square error between the input and the reconstructed patterns and maximizes the expected sparsity measure of the weight vector subject to the constraint that norm of each basis function is unity. It can be mathematically formulated as,

$$\boldsymbol{\phi^*} = \arg\min_{\{\boldsymbol{\phi_i}\}} E\left[C^*\right];\ s.t\left\|\boldsymbol{\phi_i}\right\|_2 = 1, \forall i \in 1, 2, ..., m. \tag{2}$$

where the expectation $E[.]$ is over the distribution of the input patterns. The optimal cost C^* associated with an input pattern $\boldsymbol{s}$ for a fixed basis $\{\boldsymbol{\phi_i}\}$ is given by,

$$\begin{aligned} C^* &= \min_{\{\alpha_i\}} C,\ where \\ C &= \left\| \mathbf{s} - \sum_{i=1}^{m} \alpha_i \boldsymbol{\phi_i} \right\|_2^2 + \lambda \sum_{i=1}^{m} \log\left(1 + \left(\frac{\alpha_i}{\sigma}\right)^2\right) \end{aligned} \tag{3}$$

Note that (3) has two terms. Minimizing its first term minimizes the squared error, while that of the second term maximizes the sparsity of the weight vector. Also, λ is a positive constant which controls the importance of the second term relative to the first term. Whereas σ is a constant scaling factor which is set to the standard deviation of the input patterns.

The learning of basis functions is carried out in two steps. First, we treat the basis functions as fixed and find the weights corresponding to an input pattern by solving for C^*. Second, we update the basis functions to further minimize the cost C by fixing the weights found in the first step. This procedure is repeated for all the input patterns in the training set over several epochs.

2.1. Updating the weights

For a fixed basis set and an input pattern, the optimal weights are obtained by solving a set of partial derivatives $\frac{\partial C}{\partial \alpha_i}$ being set to zero. This requires finding a solution to a set of non-linear equations. We use the Newton-Raphson technique to update the weights,

$$\alpha_i^{k+1} = \alpha_i^k + \Delta\alpha_i,\ \forall i \in 1, 2, ..., m. \tag{4}$$

In (4), $\Delta\alpha_i$ are obtained by solving the following set of linear equations[2].

$$\begin{bmatrix} \frac{\partial f_1}{\partial \alpha_1} & \frac{\partial f_1}{\partial \alpha_2} & . & . & . & \frac{\partial f_1}{\partial \alpha_m} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ \frac{\partial f_m}{\partial \alpha_1} & \frac{\partial f_m}{\partial \alpha_2} & . & . & . & \frac{\partial f_m}{\partial \alpha_m} \end{bmatrix}_{\{\alpha_i^k\}} \begin{bmatrix} \Delta\alpha_1 \\ . \\ . \\ \Delta\alpha_m \end{bmatrix} = - \begin{bmatrix} f_1 \\ . \\ . \\ f_m \end{bmatrix}_{\{\alpha_i^k\}}$$

where $f_j = -\frac{1}{2}\frac{\partial C}{\partial \alpha_j}$, and is given by

$$f_j = \langle \boldsymbol{s}, \boldsymbol{\phi_j} \rangle - \sum_{i=1}^{m} \langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j} \rangle \alpha_i - \frac{\lambda}{\sigma} \left(\frac{\frac{\alpha_j}{\sigma}}{1 + \left(\frac{\alpha_j}{\sigma}\right)^2} \right),$$

$\forall j \in 1, 2, ..., m$, and $\langle . \rangle$ indicates the inner product. Furthermore, the partial differential $\frac{\partial f_j}{\partial \alpha_i}$ can be expressed as,

$$\begin{aligned} \frac{\partial f_j}{\partial \alpha_i} &= -\langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j} \rangle,\ i \neq j. \\ &= -\langle \boldsymbol{\phi_j}, \boldsymbol{\phi_j} \rangle - \frac{\lambda}{\sigma^2} \left(\frac{1 - \left(\frac{\alpha_j}{\sigma}\right)^2}{\left(1 + \left(\frac{\alpha_j}{\sigma}\right)^2\right)^2} \right),\ i = j. \end{aligned}$$

2.2. Updating the basis functions

Gradient descent technique is applied for updating the basis functions $\boldsymbol{\phi_i}$. In this step, the weights obtained in section 2.1 are used and considered as fixed for a given input pattern. The updated basis functions $\boldsymbol{\phi_i}'$ are given by,

$$\begin{aligned} \boldsymbol{\phi_i}' &= \boldsymbol{\phi_i} - \frac{\eta}{2}\left(\frac{\partial C}{\partial \boldsymbol{\phi_i}}\right), \\ &= \boldsymbol{\phi_i} - \eta\left(-\alpha_i\left[\boldsymbol{s} - \sum_{j=1}^{m} \alpha_j \boldsymbol{\phi_j}\right]\right), \forall i \in 1, 2, ..., m. \end{aligned}$$

Where the learning parameter η is initially kept high, and its value is gradually decreased as a function of the number of epochs. The updated basis functions are normalized such that they are of unit norm.

3. OBTAINING SPARSE FEATURES

Having identified the overcomplete set of basis functions, the next important question is to how to express a given input pattern as a linear combination of these basis functions such that the representation is as sparse as possible. The weights

[2]The matrix entries $\frac{\partial f_i}{\partial \alpha_j}$ and f_i are evaluated using the weights of the k^{th} iteration $\alpha_1^k, \alpha_2^k, ..., \alpha_m^k$. The initial estimate of α_j^0 is set to be $\frac{|\boldsymbol{s}|}{m}\langle \boldsymbol{s}, \boldsymbol{\phi_j} \rangle$. Where $|.|$ indicates the cardinality and $\langle . \rangle$ represents the inner product.

of the linear combination are used as features for representing the input pattern. Compressive sampling (CS) theory exactly addresses this problem when reconstructing an input signal from its partial observations [8, 9].

Let $\mathbf{\Phi}$ be the the $n \times m$ matrix (where input dimensionality is indicated by n) which represents an overcomplete set of basis functions or a dictionary learned in section 2 *i.e.,*

$$\mathbf{\Phi} = \begin{bmatrix} \boldsymbol{\phi_1} & \boldsymbol{\phi_2} & . & . & . & \boldsymbol{\phi_m} \end{bmatrix}.$$

Then according to CS theory, the problem of determining the sparse weight vector $\boldsymbol{\alpha}$, whose elements are α_i, corresponding to an input pattern $\boldsymbol{s}$ can be posed as,

$$\arg \min_{\boldsymbol{\alpha} \in \boldsymbol{R^m}} \|\boldsymbol{\alpha}\|_{l1} \quad s.t. \quad \boldsymbol{s} = \mathbf{\Phi}\boldsymbol{\alpha}.$$

This is a linear programming problem which can be efficiently solved by many existing algorithms. In our experiments, l_1-MAGIC package is used [10].

4. RESULTS

Speaker independent phoneme recognition experiments are conducted on TIMIT in order to test the effectiveness of the proposed feature extraction technique. As mentioned earlier, our approach operates in the spectro-temporal speech domain (log critical band energies) which is obtained by first performing a Short Time Fourier Transform (STFT) with an analysis window of length 25 ms and a frameshift of 10 ms on the input speech signal. Log critical band energies are subsequently obtained by projecting the magnitude square values of the STFT output on a set of frequency weights, which are equally spaced on the Bark frequency scale, and then applying a logarithm on the output projections.

The input spectro-temporal patterns for learning the overcomplete set of basis functions are obtained from the spectro-temporal representation of the training utterances by taking a context of about 210 ms centered on each frame. The dimensionality of any such pattern (or $\boldsymbol{s}$) is 19 x 21 = 399, as there are 19 critical bands and 21 frames. Four thousand spectro-temporal patterns are randomly sampled (with uniform density) from all the patterns present in the train set in order to learn a set of $m = 429$ basis functions $\{\boldsymbol{\phi_i}\}$. All the basis functions are initialized with zero mean Gaussian White Noise (GWN) and normalized to have unit norm. Learning is accomplished by first determining the weights corresponding to an input pattern and then updating the basis functions using found weights as described in Section 2. This procedure is repeated for all four thousand input patterns and about two hundred epochs. Once the overcomplete set is identified, the feature vector corresponding to any spectro-temporal pattern is obtained by solving the l_1 minimization problem described in Section 3. Fig. 1 shows some examples of the learned basis functions. The implementation details of the phoneme recognition system are described below.

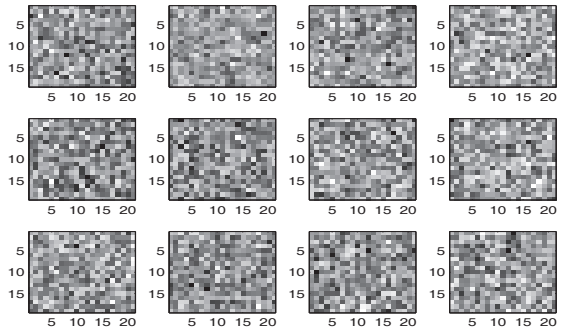

Fig. 1. Examples of sample basis functions corresponding to m = 429.

Initially, an MLP is trained to estimate the posterior probabilities of the phonemes conditioned on the input features by minimizing the cross entropy between the input acoustic feature vectors and the corresponding phoneme target classes [11]. The posterior probabilities estimated by MLP are used as the emission likelihoods (no language model) of the HMM states as described in the hybrid approach [12]. Each phoneme is modeled using 3 HMM states with equal self and transition probabilities. Decoding is accomplished by applying the Viterbi algorithm and the phoneme recognition accuracy is obtained by comparing the decoded phoneme sequence against the reference sequence. Additionally, the phoneme insertion penalty is chosen to be the one that maximizes the phoneme recognition accuracy of the CV data. Note that the silence class is ignored while evaluating the accuracies. In all of our experiments, MLP with 1000 hidden nodes is trained using the features extracted from 3000 utterances (375 speakers) of the training set and 696 utterances (87 speakers) of the cross-validation set of the TIMIT database. The test set consists of 1344 utterances of speech from 168 speakers. Furthermore, the 61 hand-labeled symbols of the TIMIT transcription are mapped to a standard set of 39 phonemes for the purpose of training and decoding [13].

The phoneme recognition accuracy of the various features is listed in Table 1. The proposed features, obtained by first learning a set of basis functions which are initialized using GWN basis and then expressing a given input pattern as a linear combination using l_1 norm minimization, perform better than the conventional PLP[3] features. It is also evident that the learning of basis is indeed useful as performance of the l_1 features with learning is better than the ones without learning. Note that the proposed features yield an absolute improvement of 0.8% over the PLP features on the (clean) TIMIT phoneme recognition task.

[3]PLP feature vector is obtained by concatenating a set of 9 frames of standard 13 PLP cepstral coefficients along with its delta and delta-delta features.

Table 1. Phoneme recognition accuracies (in %) on 16 kHz TIMIT.

Basis functions ($m = 429$)	Features	Accuracy
GWN	l_1	66.4
learned, GWN init	l_1	**67.7**
-	PLP	66.9

In order to test the noise robustness of the proposed features, we conducted phoneme recognition experiments on part of the TIMIT test set (300 randomly chosen utterances) corrupted by additive babble noise (taken from NOISEX-92) at various signal to noise ratios (SNR). The results in noisy conditions, listed in Table 2, show an average absolute improvement of 6.3% over the PLP features.

Table 2. Phoneme recognition accuracies (in %) on 16 kHz TIMIT corrupted by additive babble noise.

Features	SNR		
	10 dB	15 dB	20 dB
GWN, l_1	28.4	38.4	48.7
learned, GWN init, l_1	**30.0**	**41.3**	**51.2**
PLP	23.4	34.0	46.1

5. DISCUSSION AND FUTURE WORK

Given a set of learned basis functions (explained in section 2), in the proposed approach, features corresponding to an input pattern are obtained by minimizing the l_1 norm of the weights subject to the reconstruction of the input pattern. However, it may be interesting to see how different l_p norm minimizations for $p \geq 0$ (and their various practical implementations) perform as compared to the proposed approach.

Once trained, MLP can be viewed as a non-linear function which maps the input feature vector to the output posterior probabilities of phonemes. Ideally the posterior probability space is sparse and it contains only the linguistic (phoneme) information relevant for the task. Interestingly, the proposed feature extraction also tries to non-linearly map an input spectro-temporal pattern to a sparse feature space which preserves most of the input variability. Also by construction, the proposed features might be robust to various types of signal distortions. This is the first time the ideas of compressive sensing are applied to represent spectro-temporal patterns for speech recognition. Future work includes extensive study of the noise robustness aspect of the proposed feature extraction framework.

6. CONCLUSIONS

A novel feature extraction technique has been proposed for speech recognition based on the principles of sparse coding. We have shown how to learn the overcomplete set of basis functions from the spectro-temporal representation of speech, and how to extract features using these basis functions by solving the l_1 norm minimization problem as in the compressive sampling framework. Phoneme recognition experiments on TIMIT confirm that the proposed features perform significantly better than the conventional PLP features in both clean and noisy conditions.

7. ACKNOWLEDGEMENTS

Authors would like to acknowledge Michael Carlin, Nima Mesgarani and Sriram Ganapathy for their helpful comments.

8. REFERENCES

[1] N. Morgan et al., "Pushing the envelope-aside," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 81–88, 2005.

[2] S. Zhao and N. Morgan, "Multi-stream spectro-temporal features for robust speech recognition," in *INTERSPEECH*. Brisbane, Australia, 2008.

[3] N. Mesgarani, G.S.V.S. Sivaram, Sridhar Krishna Nemala, M. Elhilali, and H. Hermansky, "Discriminant Spectrotemporal Features for Phoneme Recognition," in *INTERSPEECH*. Brighton, 2009.

[4] D.A. Depireux, J.Z. Simon, D.J. Klein, and S.A. Shamma, "Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex," *Journal of Neurophysiology*, vol. 85, no. 3, pp. 1220–1234, 2001.

[5] B.A. Olshausen and D.J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[6] MVS Shashanka, B. Raj, and P. Smaragdis, "Sparse overcomplete decomposition for single channel speaker separation," in *ICASSP*, 2007.

[7] B.A. Olshausen and D.J Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[8] E.J. Candès and M.B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[9] D.L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[10] "l1-MAGIC," Available: http://www.acm.caltech.edu/l1magic/.

[11] M.D. Richard and R.P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural computation*, vol. 3, no. 4, pp. 461–483, 1991.

[12] H. Bourlard and N. Morgan, *Connectionist speech recognition: a hybrid approach*, Kluwer Academic Pub, 1994.

[13] K.F. Lee and H.W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.