NON-LINEAR INDEPENDENT COMPONENT ANALYSIS FOR SPEECH RECOGNITION

Mohamed Kamal Omar and Mark Hasegawa-Johnson

University of Illinois at Urbana-Champaign,
Department of Electrical And Computer Engineering,
Urbana, IL 61801. (email: omar,jhasegaw@uiuc.edu)

ABSTRACT

This paper addresses the problem of representing the speech signal using a set of features that are approximately statistically independent. This statistical independence simplifies building probabilistic models based on these features that can be used in applications like speech recognition. Since there is no evidence that the speech signal is a linear combination of separate factors or sources, we use a more general non-linear transformation of the speech signal to achieve our approximately statistically independent feature set. We choose the transformation to be symplectic to maximize the likelihood of the generated feature set. In this paper, we describe applying this nonlinear transformation to the speech time-domain data directly and to the Mel-frequency cepstrum coefficients (MFCC). The features resulted from this transformation are used in phoneme recognition experiments. The best results achieved show about 2% improvement in recognition accuracy compared to results based on MFCC features.

1. INTRODUCTION

Speech signal analysis for recognition aims at separating all information relevant for the recognition task from irrelevant information (e.g. speaker or channel characteristics) and at reducing the amount of data that is presented to the speech recognizer.

In many speech parameterization schemes, such as filter bank data obtained by sampling the short-time Fourier spectrum (STFS) nearby frequencies within the same observation are also highly correlated, inconsistent with using a diagonal covariance matrix to model conditional PDFs in hidden Markov model (HMM) recognizers. To remove some of this correlation, cepstral coefficients can be used instead of straight filter bank data. The cepstrum is obtained by taking the discrete Fourier transform (DCT) of the log of the Fourier transform of the data. The DCT approximates a Karhunen-Loève transform for a Gaussian-Markov random process. This eliminates some of the correlation between the individual parameters of a single observation frame, fitting the data more closely to the diagonal covariance assumption.

No matter what analysis method is used, the speech data is highly correlated between adjacent frames. To account for this correlation in the context of an HMM and its assumption of independence between observations, first, and sometimes higher-order derivatives, are often included in the speech parameterization. These types of additions have been shown to give improved recognition over baseline models. However, this improved performance comes at the expense of violating the diagonal covariance assumption by using features that are explicit functions of other features in the same feature vector.

The diagonal covariance mixture of Gaussians probabilistic model of the speech signal is the correct model only if conditionally independent components of speech are used as the input features of the recognizer. The conditioning is on the state level in case of using a single Gaussian for each state, and on the Gaussian component level in the case of a mixture Gaussians model. A mixture of diagonal covariance Gaussians is able to represent some correlation among the measurement dimensions, but the flexibility of a mixture Gaussian model is limited by the number of mixtures.

In this work, independent component analysis (ICA) is used to extract the independent components of speech signal. We argue that these components will be better approximated by diagonal covariance Gaussian mixture models than the acoustic features currently used in speech recognition. We describe an algorithm that separates components that are non-linearly combined together. Our algorithm does not have the limitation of most independent component analysis algorithms that prior information about the component probability density functions has to be known. Due to the symplectic property of the transformation, the output components are the maximum likelihood solution of the problem of finding the deterministic transformation of the input data to components that are modeled by a given joint PDF model that assumes independence of these components. We test using the coefficients generated by our algorithm in recognition on the TIMIT speech database.

The organization of this paper is as follows. In section 2,

our independent factor analysis approach based on nonlinear symplectic transformation is described. The application of this transformation to the speech signal is illustrated in section 3. In section 4, experiments on phoneme recognition using the TIMIT speech database are presented. Finally, the conclusion and future work are described in section 5. In this paper, a subscript is used as an index of a component of a random vector, and a superscript is used as an index of a realization of the random vector. Capital letters are used to denote the random variables and the corresponding small letters to denote their realizations.

2. NONLINEAR INDEPENDENT COMPONENT ANALYSIS

ICA algorithms assume that the components are mixed linearly to generate the observation data [1]. However, in many interesting applications, this assumption is unjustified or unacceptable. An example is the time-domain speech signal that has some components that are additively combined like voicing, aspiration, and frication sources and others are nonlinearly combined together like excitation source and vocal tract filter information that are convolutionally combined. In the cepstral domain, coarticulation effects and additive noise are examples of independent sources in the speech signal that are nonlinearly combined with the information about the vocal tract shape that is important for recognition.

In this section, an extension of the ICA algorithms to nonlinearly mixed sources is introduced. Our goal now is to find the mixing functions and the independent components given the observations. Since the components are statistically independent, we have to find the solution that minimizes the mutual information of the output components, $I(Y)$ [2]. However, to have a well-defined optimization problem, we need some restrictions on the nonlinear function or a criterion that the solution should optimize.

2.1. Problem Formulation

The mutual information is a function of the output differential entropy,

$$I(Y) = \sum_{i=1}^{n} H(Y_i) - H(Y), \tag{1}$$

where n is the number of components of the output vector, and Y_i is the ith component of the vector Y.

For a continuous random vector $Y \in \Re^n$, the mutual information is invariant to scaling but differential entropy is sensitive to it. To avoid this scale-sensitivity problem, and the need of having an estimate of the joint probability density function to calculate the differential entropy of the output vector, we choose to keep the output differential entropy equal to the input differential entropy, $H(Y) = H(X)$, while minimizing $\sum_{i=1}^{n} H(Y_i)$ to minimize the mutual information of the output vector.

It will be shown also that this choice leads to minimizing the negative of the empirical function used in maximizing the likelihood of the output vectors. This means that this approach produces a maximum likelihood transform under the constraint of the output components independence.

The relation between the output differential entropy and the input differential entropy is in general [3],

$$H(Y) \leq H(X) + \int_{\Re^n} P(x) \ln \left(det \left(\frac{\partial f(x)}{\partial x} \right) \right) dx, \tag{2}$$

where $P(x)$ is the probability density function of the random vector X, for an arbitrary transformation, $y = f(x)$, of the random vector X in $\Re^n$, with equality if $f(x)$ is invertible. For the input and output differential entropy to be equal, $f(x)$ must be invertible, and

$$det \left(\frac{\partial f}{\partial x} \right) = 1. \tag{3}$$

Equation 3 is satisfied by any volume-preserving map. Symplectic maps are a class of volume-preserving maps with useful properties [4]. An interesting property of any non-reflecting symplectic transformation from x to y, is that it can be represented using a scalar function $g(.)$ such that [5],

$$y = x - J^{-1} \frac{\partial}{\partial u} g(u); \tag{4}$$

$$u = \frac{x+y}{2};$$

$$J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix};$$

$$J = -J^{-1} \tag{5}$$

where I denotes the identity matrix in $\Re^{n/2}$. The gradient is to be taken with respect to the argument u. Now the nonlinear ICA problem can be formulated as the problem of finding the function $g(u)$ that minimizes $\sum_{i=1}^{n} H(Y_i)$ under the constraint that $H(Y) = H(X)$ guaranteed by the symplectic map. The minimum of this sum , $H(X)$, is independent of the symplectic map parameters, as for any random vector Y,

$$\sum_{i=1}^{n} H(Y_i) \geq H(Y). \tag{6}$$

We use a multi-layer feed-forward neural network to get a good approximation of the scalar function $g(u)$ [6]. The parameters of this network are optimized to minimize $\sum_{i=1}^{n} H(Y_i)$ under the constraint $H(Y) = H(X)$.

$$\hat{W} = \arg\min_{W} \sum_{i=1}^{n} H(Y_i) \tag{7}$$

where

$$W = (A, B),$$

$$g(u, A, B) = \sum_{j=1}^{M} b_j S(a_j u) \tag{8}$$

where $S(.)$ is a nonlinear function like sigmoid or hyperbolic tangent, a_j is the jth row of the $M \times n$ matrix A, and b_j is the jth element of the $M \times 1$ vector B. The constant offset term that is usually used was omitted to have a zero input zero output map.

2.2. Efficient Estimation of The Objective Function

The objective function to be minimized is

$$V = \sum_{i=1}^{n} H(Y_i). \tag{9}$$

The differential entropy of a random variable is by definition the negative of the expectation of the logarithm of its probability density function

$$H(Y_i) = -E[\log P(Y_i)], \text{ for } i = 1, 2, \cdots, n. \tag{10}$$

Since we do not have the true probability density function of the random variable Y_i, and all we can calculate is a finite set of realizations of this random variable $\{y_i^1, y_i^2, \cdots, y_i^N\}$ of size N, the expectation will be approximated by the sample mean of the given values of the random vector.

This gives the empirical estimate of the objective function as

$$V_{emp} = -\sum_{i=1}^{N} \sum_{j=1}^{n} \log P(Y_j = y_j^i), \tag{11}$$

where N is the number of samples used to estimate V_{emp}. Again, we do not have the true probability density functions of each component, therefore we use a maximum likelihood parameterized estimate of these probability density functions.

This gives the final form of the empirical estimate of the objective function as

$$V_{emp} = -\sum_{i=1}^{N} \sum_{j=1}^{n} \log P_{\Lambda_j}(Y_j = y_j^i), \tag{12}$$

where $P_{\Lambda_j}(Y_j)$ is the parameterized estimate of $P(Y_j)$ defined by the parameters Λ_j for $j = 1, 2, \cdots, n$.

Minimizing this expression is equivalent to maximizing the estimated log likelihood of the output vectors, under the assumption that the features are independent. This means that this approach can be considered as a generalization of maximum likelihood approaches to ICA to the nonlinear mixing case. Maximum likelihood approaches to ICA are closely related to the MLLT introduced in [7]. The difference is mainly in replacing the output coefficients' independence constraint of ICA by the diagonal covariance constraint of MLLT.

To calculate the gradient of the objective function with respect to the symplectic transformation parameters, we need to calculate its derivative with respect to each parameter. In general,

$$\frac{\partial V_{emp}}{\partial a_{qr}} = -\sum_{i=1}^{N} \sum_{j=1}^{n} \frac{\partial P_{\Lambda_j}(Y_j = y_j)}{\partial y_j} \frac{\partial y_j}{\partial a_{qr}} (\log P_{\Lambda_j}(Y_j = y_j) + 1)|_{y_j = y_j^i}, \tag{13}$$

$$\frac{\partial V_{emp}}{\partial b_q} = -\sum_{i=1}^{N} \sum_{j=1}^{n} \frac{\partial P_{\Lambda_j}(Y_j = y_j)}{\partial y_j} \frac{\partial y_j}{\partial b_q} (\log P_{\Lambda_j}(Y_j = y_j) + 1)|_{y_j = y_j^i}, \tag{14}$$

for

$$q = 1, 2, \cdots, M,$$

and

$$r = 1, 2, \cdots, n$$

$$\frac{\partial y_j}{\partial a_{qr}} = -h(j) \frac{\partial^2 g(u)}{\partial u_{j+h(j)\frac{n}{2}} \partial a_{qr}}, \tag{15}$$

$$\frac{\partial y_j}{\partial b_q} = -h(j) \frac{\partial^2 g(u)}{\partial u_{j+h(j)\frac{n}{2}} \partial b_q}, \tag{16}$$

$$h(j) = \begin{cases} -1 & \text{if } j \geq \frac{n}{2} \\ 1 & \text{if } j < \frac{n}{2} \end{cases}$$

3. IMPLEMENTATION OF THE ALGORITHM FOR SPEECH PROCESSING

Initially both the values of the symplectic map parameters, W, and the output vectors, y, are unknown, so we choose an initial value of the symplectic map parameters, then we solve the symplectic map equation for the output vectors. Given the output vectors corresponding to the input data, we use the expectation maximization algorithm to calculate the parameters of the probabilistic model. Based on this model, the empirical objective function is estimated, and the symplectic map parameters are updated using a conjugate gradient based method. This sequence is repeated until a local minimum of the empirical estimate of the objective function is achieved.

3.1. Estimation of The Output Vectors

To solve the symplectic transformation relation for the output vector given the input vector and the symplectic map parameters, the problem is formulated as an optimization problem. The output of the symplectic mapping is calculated using the conjugate gradient algorithm. The output vector y is the one that achieves the unconstrained minimum of

$$L(y) = \left\| y - x + J^{-1}\nabla g\left(\frac{x+y}{2}\right)\right\|^2. \quad (17)$$

The updating rule at each iteration is

$$y^{k+1} = y^k + \alpha^k d^k. \quad (18)$$

The directions of the conjugate gradient algorithm are generated by

$$d^0 = -\nabla L(y^0), \quad (19)$$
$$d^k = -\nabla L(y^k) + \zeta^k d^{k-1}, \quad (20)$$

where ζ^k is given by

$$\zeta^k = \frac{\nabla L(y^k)^T \nabla L(y^k)}{\nabla L(y^{k-1})^T \nabla L(y^{k-1})}. \quad (21)$$

The scaling factor, α^k, of the direction in each iteration is selected based on limited minimization rule on the interval $[0, h]$

$$L(y^k + \alpha^k d^k) = \min_{\alpha\in[0,h]} L(y^k + \alpha d^k), \quad (22)$$

using the golden-section search method. The algorithm is guaranteed to converge to a local minimum like all gradient-based optimization algorithms; because $L(y)$ is in general not a convex function of y, convergence to a global minimum is not guaranteed. In practice, for about 90% of the input vectors, the algorithm converged in less than 5 iterations to a value of $L(y)$ less than 0.0001.

The computational complexity of the algorithm for updating the output vectors in each iteration is $O((n + (n + 1)M)N)$, where n is the input vector length, M is the number of hidden nodes in the neural network, and N is the number of input vectors.

3.2. Evaluation of The Parameters of The Symplectic Map

After calculating the output vectors corresponding to the initial map parameters, we use the conjugate gradient algorithm to find the set of the mapping parameters that minimize minimize the regularized objective function E_g.

To be able to calculate the differential entropy of each component of the output y and its gradient, we have to define a parametric form of the PDF of the output components. In our experiments, we used the mixture of Gaussians probabilistic model for each component. In the case of the mixture Gaussian probabilistic model, the derivative of the probability density function is

$$\frac{\partial P(y_j)}{\partial y_j} = \sum_{k=1}^{K} -H_{jk}\frac{1}{\sqrt{2\pi}\sigma_{jk}}\frac{(y_j-\mu_{jk})}{\sigma_{jk}^2}\exp\left(-\frac{(y_j-\mu_k)^2}{2\sigma_k^2}\right), \quad (23)$$

$$\sum_{k=1}^{K} H_{jk} = 1$$

for all $j = 1, 2, \cdots, n$, where H_{jk} is the weight of the kth Gaussian PDF in the mixture of Gaussians, K is the number of Gaussian PDFs in the mixture of Gaussians, μ_{jk} is the mean of the kth Gaussian PDF in the mixture, and σ_{jk}^2 is the variance of the kth Gaussian PDF in the mixture. The parameters of these probabilistic models are calculated from the output data using the expectation-maximization (EM) algorithm [8]. We used the hyperbolic tangent function as the nonlinear function in the feed forward neural network approximation of the scalar function that is used in the symplectic map.

Therefore, the derivatives of the output components with respect to the symplectic map parameters become

$$\frac{\partial y_j}{\partial a_{qr}} = \begin{cases} 2h(j)b_q a_{qj+h(j)\frac{n}{2}} g(a_q y)(1-g^2(a_q y))\frac{x_r+y_r}{2} \\ \quad \text{if } r \neq j + h(j)\frac{n}{2} \\ 2h(j)b_q a_{qj+h(j)\frac{n}{2}} g(a_q y)(1-g^2(a_q y))\frac{x_r+y_r}{2} \\ -h(j)b_q(1-g^2(a_q y)) \quad \text{if } r = j + h(j)\frac{n}{2} \end{cases} \quad (24)$$

$$\frac{\partial y_j}{\partial b_q} = -h(j)a_{qj+h(j)\frac{n}{2}}(1-g^2(a_q y)). \quad (25)$$

Substituting the derivatives of the output components with respect to the symplectic map parameters and the derivatives of the probability density function with respect to the output components for both parametric PDF forms in equations 13, 14, 15, and 16, we get the derivatives of the empirical objective function with respect to the symplectic parameters. Adding to these derivatives, the derivatives of the regularization term, we get the derivatives of the regularized objective function with respect to the symplectic parameters. Given these derivatives, we use any the conjugate gradient algorithm to update the values of the symplectic parameters. The computational complexity of the algorithm for updating the symplectic parameters in each iteration is $O((3nK + (n+1)M + n^2M)N)$, where n is the input vector length, M is the number of hidden nodes in the neural network, K is the number of Gaussian components in the mixture, and N is the number of input vectors.

In the next section, we will provide experiments that used these implementations.

4. EXPERIMENTS AND RESULTS

Our approach to nonlinear ICA was applied to the speech signal. First, it was applied to the speech samples directly in the time domain, and then it was applied to the MFCC coefficients in the Cepstrum domain.

In the direct time domain processing, the TIMIT speech database, with sampling rate at 16 KHZ, is downsampled to 8 KHZ and preemphasized. Each utterance of speech is divided to fixed-size frames of length 20 samples. Then 1000 of these frames are used at a time to update the values of the parameters of the symplectic transformation and the marginal probability density functions of the output components.

In the cepstral domain processing, the overall feature vector consists of 12 MFCC coefficients in the first two experiments in cepstral domain. The last experiment uses 12 MFCC coefficients, energy and their deltas. In both cases, this MFCC based feature vector is used as the input to our symplectic non-linear independent component analysis.

In each iteration, the output components are calculated using the current symplectic transformation parameters by using the symplectic mapping equation, then the maximum likelihood estimates of the marginal probability density functions of the output components are calculated using the EM algorithm. Then, the sum of the differential entropy of the output components is calculated and its gradient and the symplectic mapping parameters are updated such that this sum is minimized. After the iterative algorithm converges to a set of locally optimal symplectic parameters, the training data are transformed by the symplectic map yielding corresponding output coefficients. The output coefficients are compared to linear ICA, and MLLT in their recognition accuracy.

The 61 phonemes defined in the TIMIT database are mapped to 48 phoneme labels for each frame of speech as described in [9]. These 48 phonemes are collapsed to 39 phoneme for testing purposes as in [9]. A three-state left-to-right model for each triphone is trained using the EM algorithm. The number of mixtures per state was fixed to five. After training the overall system and obtaining the symplectic map parameters, the approximately independent output coefficients of the symplectic map are used as the input acoustic features to a Gaussian mixture hidden Markov model speech recognizer [9]. The parameters of the recognizer are trained using the training portion of the TIMIT database.

To compare the performance of the proposed algorithm with other approaches, we generated acoustic features using linear ICA, and MLLT. We used the maximum likelihood approach to linear ICA as described in [1] and briefly overviewed in section 2. Finally we implemented MLLT as described in [7]. All these techniques used a feature vector that consists of twelve MFCC coefficients, the energy, and their deltas as their input.

Testing this recognizer, using the test data in the TIMIT database, we get the phoneme recognition results in table 1. These results are obtained by using a bigram phoneme language model and by keeping the insertion error around 10% as in [9]. The table compares these recognition results to the ones obtained by MFCC, linear ICA, and MLLT.

To improve the phoneme recognition accuracy in the time domain, we used a linear map that maximizes an empirical estimate of the mutual information between the phoneme identities and the output coefficients [10]. These linear maps were used on each component separately and therefore preserved the approximate independence property of the components generated by the nonlinear symplectic map. Using these features, generated by trying to maximize the mutual information, we trained the previously described HMM recognizer. As shown in table 1, the phoneme recognition accuracy is improved by using this linear map, but still it falls behind the phoneme recognition accuracy achieved by MFCC acoustic features.

These results encouraged us to perform the nonlinear independent component analysis on the MFCC coefficients instead of the time-domain signal directly.

Three different kinds of experiments were done to test the phoneme recognition results based on the nonlinear ICA coefficients generated with MFCC inputs. First, the twelve cepstrum coefficients were used as the input vector to the nonlinear component analysis algorithm, and the energy was added to the output coefficients. The resultant 13-coefficient feature vector was used to train the HMM recognizer. In the second experiment, we added the delta of the output coefficients and the energy to the acoustic vector that is used in the first experiment. The resultant 26-coefficient feature vectors were used to train the HMM recognizer. Finally, we used the twelve cepstrum coefficients, the energy, and their deltas as the input to the nonlinear independent component analysis algorithm, and used the 26 output coefficients as the acoustic vector that is used in phoneme recognition. As shown in table 1, the best results were achieved by using the cepstrum coefficients, the energy, and their deltas as the input to the nonlinear symplectic map and using the output of the map as the acoustic feature vector for the phoneme recognizer.

Comparing the phoneme recognition results of the symplectic map in the cepstral domain to the results obtained using the symplectic map on the time-domain data, we find that the features obtained from the mapping of the MFCC features outperform those obtained from the time-domain data. Also, adding the delta coefficients to the MFCC co-

efficients increases the phoneme recognition accuracy by about 7%. As shown in table 1, the MLLT performed the best among linear transforms with about 0.9% improvement over the MFCC-based feature vector. Comparing these results with the non-linear ICA algorithm in the cepstral domain, we find that non-linear ICA outperforms the best linear approach by 1% using the same length of the features vector.

Table 1. Phoneme Recognition Accuracy

Acoustic Features	**Recognition Accuracy**
MFCC	73.7%
Linear ICA	73.5%
MLLT	74.6%
Non-Linear ICA (NICA) in Time Domain	61.2%
NICA in Time Domain After MMI Mapping	64.4%
NICA(Static MFCC) +Energy	68.7%
NICA(Static MFCC) +△NICA+Energy+△Energy	71.2%
NICA(Static MFCC +Energy+△MFCC+△Energy)	75.6%

5. DISCUSSION

In this work, we introduced a nonlinear symplectic independent component analysis algorithm. This algorithm can provide the maximum likelihood transform of the features under the independece constraint on the transformed features. This algorithm was applied to the speech signal in two different ways. First, it was applied to the time-domain speech data and the output coefficients' phoneme recognition accuracy were evaluated. Second, we applied our algorithm to the MFCC features of the speech signal and its energy. We compared the phoneme recognition accuracy of the output coefficients to linear ICA, and MLLT. In this case, the phoneme recognition accuracy is improved compared to MFCC, linear ICA, and MLLT. The best phoneme recognition accuracy is achieved when the MFCC, energy and their deltas are used as input to the nonlinear ICA algorithm. This can be attributed to the ability of the algorithm to find a better representation of the acoustic clues of different phonemes when provided with input features that have proved to be efficient in coding the acoustic information that is related to phonemes. The improvement due to this different representation over the input MFCC features that have the same amount of information about phonemes, is due to the approximate independence property of the new features that allow a more efficient probabilistic modeling of the conditional probabilities with the same model complexity.

6. ACKNOWLEDGMENT

This work was supported by NSF award number 0132900.

7. REFERENCES

[1] Te-Won Lee, Mark Girolami, Anthony J. Bell, and Terrence J. Sejnowski, "A Unifying Information-Theoretic Framework For Independent Component Analysis, " *Computers & Mathematics with Applications* , Vol. 31 (11), pp. 1-21, March 2000.

[2] Thomas M. Cover, and Joy A. Thomas, *Elements of Information Theory*, Wiley, New York, NY, 1997.

[3] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1991.

[4] Ralph Abraham, Jerrold E. Masden, *Foundation of Mechanics*, The Benjamin/Cummings Publishing Company, 1978.

[5] L. Parra, G. Deco, S. Miesbach, "Statistical independence and Novelty Detection with Information Preserving Nonlinear Maps, " *Neural Computation*, 8, pp. 260-269, 1996.

[6] K. Hornik, M. Stinchcombe, H. White, "Multilayer Feed-forward Neural Networks Are Universal Approximators, " *Neural Network*, 2, pp. 359-366, 1989.

[7] R. A. Gopinath, "Maximum Likelihood Modelling With Gaussian Distributions For Classification," *IEEE Proceedings of ICASSP*, Seattle, Washington, 1998.

[8] Todd K. Moon, "The Expectation Maximization Algorithm," *IEEE Signal Processing Magazine*, pp. 47-60, November 1996.

[9] Kai-Fu Lee, and Hsiao-Wuen Hon, "Speaker Independent Phone Recognition Using Hidden Markov Models," *IEEE Trans. on ASSP*, 37(11), pp. 1641-1648, November 1989.

[10] Mohamed Kamal Omar, and Mark Hasegawa-Johnson, "Maximum Mutual Information Based Acoustic-Features Representation of Phonological Features For Speech Recognition," *IEEE Proceedings of ICASSP*, Orlando, Florida, 2002.