1

Approximately Independent Factors of Speech Using Non-Linear Symplectic Transformation

Mohamed Kamal Omar and Mark Hasegawa-Johnson

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). We discuss also experiments in which the generated feature set is transformed into a more compact set using a maximum mutual information linear transformation. This linear transformation is used to generate the acoustic features that represent the distinctions among the phonemes. 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.**

I. INTRODUCTION

The purpose of this work is to construct approximately independent basis for the speech signal. Starting from early researchers of audio perception, like Helmholtz [1], it was observed that the environment that we live in has been a major shaping factor to the development of our perception. Within a decade of the development of information theory by Shannon [2], its concepts were used to express the perceptual processes. Attneave observed that perceptually important information on natural images are the edges of objects in the image [3]. Barlow explored a possible structure of our neural system to perform such optimizations [4]. He suggested that the sensory mechanism performs sparse or factorial coding [5], [6]. Sparse coding is the decomposition of the signal that results in compression of signal energy into the smallest possible number of coefficients, and factorial coding is the decomposition that strives for the coefficients that are as mutually independent as possible. Bell and Sejnowski used modern information theory optimization algorithms to sparsely analyze natural images [7], [8]. They developed an algorithm for separating the statistically independent components of a data set through unsupervised learning. The algorithm belongs to the independent component analysis algorithms. These algorithms can be considered as a generalization of the principal component algorithm in order to separate the higher-order dependencies in the input, in addition to the second-order dependencies. Independent component analysis has an inherent assumption that the independent components are combined linearly, and hence can be separated by linear transformation.

Physically all sounds including the speech signal are a series of pressure changes in the medium between the sound source and the listener. The objective of signal analysis is to produce a parameterization of the speech signal suitable for automatic speech recognition or coding. Speech analysis for recognition aims at separating information relevant for the recognition task from irrelevant information (e.g. speaker or channel characteristics), reducing the amount of data that is presented to the speech recognizer, and satisfying the constraints imposed by the speech recognizer on the speech representation.

Speech data can be parameterized in many different ways. The two main approaches are some type of coding — usually linear prediction— of the time domain, and direct sampling of domains other than the time domain, usually the frequency or cepstral domains. Irino and Kawahari noted that the function of the basilar membrane is similar to a wavelet transform of the signal, and developed a special class of wavelets known as the "auditory wavelet transform" based on the impulse response of the cochlea at 1400 Hz [9]. In all of these approaches, the input speech samples are windowed and the resulting speech segments termed frames. The data analysis is then executed on each frame, which corresponds to single observations with regard to a hidden Markov model (HMM).

In many coding 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 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 cosine 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. Experiments have also been done where the parameters of the previous frame are included in the current frame, also trying to account for the correlation between adjacent vectors [10]. These types of additions have been shown to give improved recognition over baseline models, possibly indicating a better match

between the speech data and the HMM assumption. 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.

Representing the speech signal in terms of independent components will result in a better satisfaction of the diagonal covariance assumption, a common assumption in practical Gaussian mixture HMMs. The Gaussian mixture HMM is the most successful speech recognition model up to now [11], [12], but using the full covariance matrix is impractical due to the high dimensionality of the 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.

In this work, we introduce a generalization of the independent component analysis approach. 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 apply our nonlinear ICA approach to the speech signal. In this application, a best basis of the speech signal is selected by finding its independent components. We test using the coefficients generated by our algorithm in recognition on the TIMIT speech database. This application of ICA is an attempt to design a flexible signal processing technique tailored for the speech signal that competes with the current popular time-frequency analysis techniques. It provides basis signals that are learned from the training data directly. This approach to speech signal analysis may decrease the gap between the properties of the speech signal representation and the assumptions made about it in probabilistic models of speech used in recognition systems like hidden Markov model (HMM) [13] [14].

The organization of this paper is as follows. Section 2 discusses principal component analysis and its extension to independent component analysis. In section 3, 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 4. In section 5, experiments on phoneme recognition using the TIMIT speech database are presented. Finally, the conclusion and future work are described in section 6. 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.

II. Component Analysis

The first stage in many pattern recognition and coding tasks is to generate a good set of features from the observed data. The set should be compact and capture all class discriminating information in case of recognition and all information needed to reconstruct the observed data with sufficient quality in case of coding. Features that contain little or no information should be avoided since they increase the computational load and the storage and transmission requirements without improving the performance. One of the powerful techniques for extracting structure from high-dimensional data is principal component analysis.

A. Principal Component Analysis

Principal component analysis, and the closely related Karhunen-Loève transform are classic techniques in statistical data analysis, feature extraction, and data compression [15]. Given a random vector X and a number of observations from this random vector, no explicit assumptions on the probability density of the vectors are made in PCA, as long as the first- and second-order statistics can be estimated from the observed data. Also, no generative model is assumed for the vector X, but there are extensions to PCA like probabilistic principal component analysis (PPCA) [16] that associate a generative model with PCA. In the PCA transform, the vector x of length n is first centered by subtracting its mean. Next, x is linearly transformed to another vector y with m elements, $m \leq n$, so that the redundancy induced by correlation is removed. This is done by finding a rotated orthogonal coordinate system such that the elements of x in the new coordinates become uncorrelated. The vector is projected in this new coordinate system to the subspace consisting of the directions along which the vector has maximum variance. The transform is constructed from the eigenvectors of the sample covariance matrix with maximum corresponding eigenvalues. This transform is the unique unitary transform of dimension m such that the elements of y are uncorrelated and the variance of y is maximized. PCA is a linear technique, so computing y from x is not computationally expensive, which makes real-time processing possible.

Since there are many sources of variability in speech features and some of them are irrelevant to linguistic information, selecting the direction of maximum variance for

projection does not always minimize the recognition error [17]. In order to maximize linguistic relevance, many speech recognizers use linear discriminant analysis (LDA) instead of PCA. LDA calculates the principal components of the Fisher covariance matrix of the classes corresponding to the speech units [18], [19],

$$S_{wb} = W^{-1}B, \tag{1}$$

where W is within-class scatter, and B is the between-class scatter [15].

LDA tries to improve the linear separability of the classes by finding the linear transform that maximizes the ratio of the determinant of between-class covariance and the determinant of the average within-class covariance [15]. Given a set of N independent observation vectors $\{x^i\}_{1\leq i\leq N}, x^i \in \Re^n$, each of them belongs to only one class $j \in 1, \cdots, J$. Let each class j be characterized by its mean μ_j, covariance matrix Σ_j, and observation count N_j. The within-class scatter is given by

$$W = \frac{1}{N}\sum_{j=1}^{J} N_j \Sigma_j, \tag{2}$$

and the between-class scatter is given by

$$B = \frac{1}{N}\sum_{j=1}^{J} N_j \mu_j \mu_j^T - \mu\mu^T, \tag{3}$$

where μ is the global mean of the observations. The goal of LDA is to find a linear transformation characterized by the matrix θ such that

$$J(\theta) = \frac{\left|\theta B \theta^T\right|}{\left|\theta W \theta^T\right|}, \tag{4}$$

is maximized. Maximization of Equation 4 can be formulated as PCA of Fisher covariance matrix or as a maximum likelihood estimation problem [20]. In [18], using PCA analysis had no effect on the phoneme classification accuracy on OGI numbers database. LDA improved the phoneme classification on the same task by 0.7%.

LDA assumes that all the within-class covariance matrices are approximately the same. This makes it inappropriate for problems like speech recognition, in which the classes have unequal within-class covariance matrices. Heteroscedastic LDA (HLDA) is an extension to LDA that removes this constraint [20]. In HLDA, the objective function to be maximized is

$$J(\theta) = \frac{\left|\theta B \theta^T\right|}{\prod_{j=1}^{J} \left|\theta \Sigma_j \theta^T\right|^{N_j}}. \tag{5}$$

This maximization can be formulated as a maximum likelihood estimation problem for normal populations with common covariance matrix in the rejected subspace. An alternative interpretation of HLDA as a constrained maximum likelihood projection for a full-covariance Gaussian model is introduced in [21] and called heteroscedastic discriminant analysis (HDA). A maximum likelihood linear transform (MLLT) which turns out to be special case of HLDA when the input and output dimensions are the same was introduced in [22]. In [21], HDA made no improvement in word recognition, but made a significant improvement when used in combination with MLLT. The non-linear independent component analysis introduced here can be shown to be a generalization of the MLLT to non-linear transforms. This is due to the fact that the empirical estimate of the objective function to be minimized in our work is actually the negative of the empirical estimate of the likelihood in MLLT approach. As the problem is formulated, we will show the equivalence of minimizing the mutual information of the output components and maximizing the likelihood of the outputs.

B. Independent Component Analysis

ICA defines a generative model for the observed multivariate data [23], [24]. This data is typically given as a large database of samples. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non Gaussian and mutually independent, and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA.

The goal of ICA is to estimate the independent sources and the mixing coefficients given only observations that are a linear mixture of the latent independent source signals. In contrast to PCA, ICA not only decorrelates the sources but also reduces higher-order statistical dependencies, attempting to make the components as independent as possible.

The data analyzed by ICA could originate from many different kinds of application fields, including digital images and document databases, as well as economic indicators and psychometric measurements. In many cases, the measurements are given as a set of parallel signals or time series; the term blind source separation is used to characterize this problem. Typical examples are mixtures of simultaneous speech signals that have been picked up by several microphones, brain waves recorded by multiple sensors, interfering radio signals arriving at a mobile phone, or parallel time series obtained from some industrial process.

The goal of ICA algorithms is to find the linear transformation W of the dependent observation vector X that makes the outputs as statistically independent as possible. This means to minimize the mutual information of the output vector Y, since

$$I(Y) \geq 0,$$

with equality if and only if the output vector components are statistically independent.

There are many approaches to solving the ICA problem, including the information maximization approach, Maximum likelihood estimation, Negentropy maximization, Higher-order moments and cumulants approxima-

tions of differential entropy, and nonlinear PCA. In [25], it is shown that all these different approaches lead to the same iterative learning algorithm.

ICA has been used in speech recognition applications when there is a background auditory source other than the speaker, [26], [27]. It was used also in developing features for speaker recognition [28], and speech recognition [29], [30], [31]. Factor analysis also was used to model the covariance matrix of the Gaussian mixtures of HMM recognizers in [32].

III. Nonlinear Independent Component Analysis

As stated in the previous section, ICA algorithms assume that the components are mixed linearly to generate the observation data. 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. The source-filter model proposes that the excitation signal and the vocal tract filter are linearly combined in the cepstral domain, but the source-filter model is unrealistic in many cases, especially for consonants. Time-varying filters and filter-dependent sources result in nonlinear source-filter combination in the cepstral domain [33].

In this paper, 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, $I(Y)$ [34]. However, to have a well-defined optimization problem, we need some restrictions on the nonlinear function or a criterion that the solution should optimize.

A. Problem Formulation

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

$$I(Y) = \sum_{i=1}^{n} H(Y_i) - H(Y), \tag{6}$$

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 [35],

$$H(Y) \;\leq\; H(X) + \int_{\Re^n} P(x) \ln \left(det \left(\frac{\partial f(x)}{\partial x} \right) \right) dx, \tag{7}$$

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

$$det \left(\frac{\partial f}{\partial x} \right) = 1. \tag{8}$$

Equation 8 is satisfied by any volume-preserving map. Symplectic maps are a class of volume-preserving maps with useful properties [36]. 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 [37],

$$y \;=\; x - J^{-1} \frac{\partial}{\partial u} g(u); \tag{9}$$

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

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

$$J \;=\; -J^{-1} \tag{10}$$

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,

$$\sum_{i=1}^{n} H(Y_i) \;\geq\; H(Y). \tag{11}$$

We use a multi-layer feed-forward neural network to get a good approximation of the scalar function $g(u)$ [38]. The parameters of this network are optimized to minimize $\sum_{i=1}^{n} H(Y_i)$ under the constraint $H(Y) = H(X)$.

$$\hat{W} = \arg\min_{W} \sum_{i=1}^{n} H(Y_i) \tag{12}$$

where

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

$$g(u, A, B) = \sum_{j=1}^{M} b_j S(a_j u) \tag{13}$$

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.

B. Efficient Estimation of The Objective Function

The objective function to be minimized is

$$V = \sum_{i=1}^{n} H(Y_i). \tag{14}$$

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

$$H(Y_i) = -E[\log P(Y_i)], \text{ for } i = 1, 2, \cdots, n. \tag{15}$$

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 is justified by the weak law of large numbers which states that if the set $\{y^1, y^2, \cdots, y^N\}$ are independent and identically distributed then [35]

$$\frac{1}{N} \sum_{i=1}^{N} y^i \to E[Y], \text{ in probability as } N \to \infty. \tag{16}$$

This gives the empirical estimate of the objective function as

$$V_{emp} = -\sum_{i=1}^{N} \sum_{j=1}^{n} \log P(Y_j = y_j^i), \tag{17}$$

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

$$V_{emp} = -\sum_{i=1}^{N} \sum_{j=1}^{n} \log P_{\Lambda_j}(Y_j = y_j^i), \tag{18}$$

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 [22]. 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,

$$\frac{\partial V_{emp}}{\partial a_{qr}} = -\sum_{i=1}^{N} \sum_{j=1}^{n} \frac{\partial P_{\Lambda_j}(Y_j = y_j)}{\partial y_j} \frac{\partial y_j}{\partial a_{qr}} \left(\log P_{\Lambda_j}(Y_j = y_j) + 1\right)|_{y_j = y_j^i}, \tag{19}$$

$$\frac{\partial V_{emp}}{\partial b_q} = -\sum_{i=1}^{N} \sum_{j=1}^{n} \frac{\partial P_{\Lambda_j}(Y_j = y_j)}{\partial y_j} \frac{\partial y_j}{\partial b_q} \left(\log P_{\Lambda_j}(Y_j = y_j) + 1\right)|_{y_j = y_j^i}, \tag{20}$$

for

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

and

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

$$\frac{\partial y_j}{\partial a_{qr}} = -h(j) \frac{\partial^2 g(u)}{\partial u_{j+h(j)\frac{n}{2}} \partial a_{qr}}, \tag{21}$$

$$\frac{\partial y_j}{\partial b_q} = -h(j) \frac{\partial^2 g(u)}{\partial u_{j+h(j)\frac{n}{2}} \partial b_q}, \tag{22}$$

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

This formulation of the symplectic parameters evaluation as a minimization problem is ill-posed, as very small changes in the values of the parameters may lead to drastic changes in the objective function. The instability of the solution arises from the fact that the output is related to the input using an implicit form. Once the instability for a given set of data samples causes the absolute value of the symplectic parameters to be large, the output of the feed-forward network saturates and becomes less dependent on the values of the symplectic parameters. This means that the algorithm will not converge and the effect of any additional input data sets will be minimal on the final values of the symplectic parameters. To make the problem well posed, the map from X to Y should be continuous,and the map from $X \times Y$ to $g(.)$ should be continous also. If Y was represented as an explicit function of X using the feed forward neural network, the continuity of both maps will be forced by this representation. But due to the implicit function representation of the symplectic map, the value of y for a given x is estimated by an optimization problem and the map is no longer guaranteed to be continuous. The problem becomes well-posed by restricting

the set from which $g(u)$ is chosen to some compact set G. One can show by virtue of the operator inversion lemma [39], that in this case the problem of empirical risk minimization becomes well posed. One can show also that if $g(u)$ is sufficiently well-behaved, i.e, has a finite covering number, the empirical objective function will converge to the actual objective function for increasing sample size N, i.e.

$$Pr\left(\sup_{g\in G}|V[g]-V_{emp}[g]|\geq\epsilon\right)\rightarrow 0 \quad \text{for } N\rightarrow\infty \text{ and } \epsilon>0 \tag{23}$$

Vapnik and Chervonenkis show that such a condition is necessary and sufficient to give uniform convergence bounds [40]. Classical regularization theory provides a solution to this type of problem in which a function is to be approximated from sparse data [41]. It formulates the regression problem as a variational problem of finding the function $g(u)\in G$ that minimizes the functional

$$E_g \;=\; \frac{1}{N}\sum_{i=1}^{N} V_{emp}(x_i,y_i,g)+\lambda\|g\|_K^2, \tag{24}$$

where $\|g\|_K^2$ is a norm in a reproducing Kernel Hilbert Space G defined by the positive definite function K, N is the number of data samples. The functionals of classical regularization lacked a rigorous justification for a finite set of training data. Vapnik has provided a general theory that justifies regularization functionals for learning from a finite set of data [42]. In the framework of structural risk minimization (SRM) suggested by Vapnik, [42], [43] , we can define a structure using a nested sequence of hypothesis spaces $G_1\subset G_2\subset\cdots\subset G_{l(N)}$ with G_m being the set of functions $g(u)$ in the reproducing kernel Hilbert space (RKHS) with

$$\|g\|_K \;\leq\; C_m, \tag{25}$$

where C_m is a monotonically increasing sequence of positive constants. For each m, we are supposed to minimize the empirical objective function subject to this constraint. This in turn leads to using the Lagrange multiplier λ_m and to minimizing

$$\frac{1}{N}\sum_{i=1}^{N} V_{emp}(x_i,y_i,g)+\lambda_m(\|g\|_K^2-C_m^2),$$

with respect to the symplectic parameters and maximizing with respect to $\lambda_m\geq 0$. The solution of this optimization problem is the same as the solution for minimizing

$$\frac{1}{N}\sum_{i=1}^{N} V_{emp}(x_i,y_i,g)+\lambda^*(N)(\|g\|_K^2-C_m^2),$$

with respect to the symplectic maps, where $\lambda^*(N)$ is the optimal Lagrange multiplier corresponding to the optimal element of the structure $C_{l^*(N)}$.

In practice this structure is formulated by imposing a convex penalty term on some quantity, $Q(g)$, related to $g(u)$ which is not necessarily the norm of the function in the reproducing kernel Hilbert space [44]. This functional has to be convex and continuous. The value of $\lambda^*(N)$ is usually chosen from a finite set of possible values or set to a constant value.

In this work, we used the square of the ℓ_2 norm of the symplectic parameters vector, $W=(A,B)$,

$$\|W\|_2^2=\sum_{i=1}^{m}|w_i|^2, \tag{26}$$

where w_i is the ith element of the vector W, and m is the length of the vector, as the convex penalty and selected the optimal Lagrange multiplier $\lambda^*(N)$ from a finite set of ten values. The value of $C_{l^*(N)}$ also was selected from a finite set of four values.

IV. 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.

A. 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 conjugate gradient algorithm [45], is used to calculate the output vector y that achieves the unconstrained minimum of

$$L(y) \;=\; \left\|y-x+J^{-1}\nabla g\left(\frac{x+y}{2}\right)\right\|^2. \tag{27}$$

The updating rule at each iteration is

$$y^{k+1}=y^k+\alpha^k d^k. \tag{28}$$

The directions of the conjugate gradient algorithm are generated by

$$d^0 \;=\; -\nabla L(y^0), \tag{29}$$

$$d^k \;=\; -\nabla L(y^k)+\zeta^k d^{k-1}, \tag{30}$$

where ζ^k is given by

$$\zeta^k \;=\; \frac{\nabla L(y^k)^T\nabla L(y^k)}{\nabla L(y^{k-1})^T\nabla L(y^{k-1})}. \tag{31}$$

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

$$L(y^k + \alpha^k d^k) = \min_{\alpha \in [0,h]} L(y^k + \alpha d^k), \quad (32)$$

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. Before using the regularization term in the objective function, the convergence of this algorithm was slow, and it sometimes failed to converge, when the input data consisted of time-domain speech samples.

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.

B. 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 both the mixture of Gaussians and the generalized Gaussian probabilistic model for each component. The motivation of choosing these specific forms is that both are general enough to approximate any PDF from the exponential family, while the mixture of Gaussians is the better choice to approximate a multi-modal PDF. The mixture of Gaussians is usually used to model the conditional PDF of MFCC coefficients in speech recognition that is known to be multi-modal [46], while the generalized Gaussian is known to approximate well the PDF of the time-domain speech samples that is known to be unimodal [28]. In all experiments described in this work, we used both parametric forms and reported the one that gave the best results. The generalized Gaussian PDF gave better results for direct time-domain processing of the speech signal, while the mixture of Gaussians PDF gave better results for cepstral-domain experiments.

The mixture Gaussian model is given by

$$P(y_j) = \sum_{k=1}^{K} H_{jk} \frac{1}{\sqrt{2\pi}\sigma_{jk}} \exp\left(-\frac{(y_j - \mu_{jk})^2}{2\sigma_{jk}^2}\right), \quad (33)$$

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

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, and the generalized Gaussian probability distribution model for each component is

$$P(y_j) = \frac{\omega(\beta_j)}{\sigma_j} \exp\left[-c(\beta_j)\left|\frac{y_j - \mu_j}{\sigma_j}\right|^{2/(1+\beta_j)}\right], \quad (34)$$

for all $j = 1, 2, \cdots, n$, where

$$c(\beta_j) = \frac{\Gamma\left[\frac{3}{2}(1+\beta_j)\right]}{\Gamma\left[\frac{1}{2}(1+\beta_j)\right]^{1/(1+\beta_j)}}, \quad (35)$$

and

$$\omega(\beta_j) = \frac{\Gamma\left[\frac{3}{2}(1+\beta_j)\right]^{1/2}}{(1+\beta_j)\Gamma\left[\frac{1}{2}(1+\beta_j)\right]^{3/2}}, \quad (36)$$

μ_j is the mean of y_j, σ_j^2 is the variance of y_j, and β is a measure of the kurtosis and a parameter that controls the distribution's deviation from normality. In the case of the mixture Gaussian probabilistic model, the derivative of the probability density function is

$$\frac{\partial P(y_j)}{\partial y_j} = \sum_{k=1}^{K} -H_{jk} \frac{1}{\sqrt{2\pi}\sigma_{jk}} \frac{(y_j - \mu_{jk})}{\sigma_{jk}^2} \exp\left(-\frac{(y_j - \mu_k)^2}{2\sigma_k^2}\right), \quad (37)$$

and in the case of the generalized Gaussian distribution model, the derivative of the probability density function is

$$\frac{\partial P(y_j)}{\partial y_j} = -c(\beta_j)\frac{2}{1+\beta_j}\left|\frac{y_j - \mu_j}{\sigma_j}\right|^{\frac{2}{1+\beta_j}-1} P(y_j). \quad (38)$$

The parameters of these probabilistic models are calculated from the output data using the expectation-maximization (EM) algorithm [47]. 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,

$$S(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \quad (39)$$

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

$$\frac{\partial y_j}{\partial a_{qr}} = \begin{cases} 2h(j)b_q a_{qj+h(j)\frac{n}{2}} g(a_q y)(1 - g^2(a_q y))\frac{x_r + y_r}{2} \\ \quad \text{if } r \neq j + h(j)\frac{n}{2} \\ 2h(j)b_q a_{qj+h(j)\frac{n}{2}} g(a_q y)(1 - g^2(a_q y))\frac{x_r + y_r}{2} \\ -h(j)b_q(1 - g^2(a_q y)) \quad \text{if } r = j + h(j)\frac{n}{2} \end{cases} \quad (40)$$

$$\frac{\partial y_j}{\partial b_q} = -h(j)a_{qj+h(j)\frac{n}{2}}(1 - g^2(a_q y)). \quad (41)$$

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 19, 20, 21, and 22, 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 can use any gradient-based algorithm to update the values of the symplectic parameters. We chose the conjugate gradient algorithm due to its fast convergence compared to other gradient based methods. 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 $K = 1$ for generalized Gaussian PDF, and N is the number of input vectors.

In the next section, we will provide experiments that used these implementations with the mixture of Gaussians and generalized Gaussian probabilistic models.

V. 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. The time domain processing of speech has applications in speech coding, prosody recognition, and speaker recognition, while processing of MFCC can be used in speech recognition.

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 Mel-frequency Cepstrum Coefficients are calculated for 4500 utterances from the TIMIT database. 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 LPCC and MFCC coefficients in their coding efficiency, and to LDA, linear ICA, and MLLT in their recognition accuracy.

A. Coding Efficiency And Sparseness Of Output Coefficients

Coding efficiency of acoustic features that are used in speech recognition is receiving much more attention recently. This is due to the growing interest in distributed speech recognition systems especially over limited bandwidth networks like wireless networks. We used the empirical estimate of the differential entropy, V_{emp}, as a measure of the number of bits required to code each coefficient. Table I compares the empirical estimate of the differential entropy of the coefficients obtained using nonlinear ICA algorithm in the time domain and the cepstral domain to the empirical estimate of the differential entropy of MFCC and LPCC coefficients. The table shows that coefficients that are generated by nonlinear ICA can be more efficiently coded than MFCC and LPCC coefficients.

Another important feature of the output coefficients that are generated by non-linear ICA is the sparseness of the output feature set. Sparseness is related to reducing the redundancy in the representation of the input signal. Given a dictionary of basis functions $S_1(u), S_2(u), \cdots, S_m(u)$, sparse approximation techniques seek an approximation of a function $g(u)$ as a linear combination of the smallest number of elements of the dictionary, that is, an approximation of the form:

$$f_w(u) \;=\; \sum_{q=1}^{m} w_q S_q(u), \tag{42}$$

with the smallest number of non-zero coefficients w_q [48]. The problem can be formulated as minimizing the following cost function

$$E[w] \;=\; D(g(u), \sum_{q=1}^{m} w_q S_q(u)) + \epsilon \|w\|_{\ell_0}, \tag{43}$$

where D is a cost measuring the distance in some predefined norm between the true function, $g(u)$, and our approximation, the ℓ_0 norm of a vector counts the number of elements of that vector which are different from zero, and ϵ is a parameter that controls the trade off between the sparseness and the goodness of the approximation. Unfortunately, it can be shown that minimizing this cost function is NP-hard because of the ℓ_0 norm. Therefore, the ℓ_0 norm is usually approximated by some other kind of norm like the ℓ_2 norm.

In our work, we choose $g(u)$ to be the scalar function in the symplectic mapping that generates the independent components of the input data, and D is taken as the sum of the differential entropy of these output components. We

choose also to use the ℓ_2 norm. Comparing this optimization problem with the one we adopted in our algorithm, we find that they are identical and therefore our algorithm is expected to provide a relatively sparse representation of the scalar function $g(u)$. A sparse representation of $g(u)$ does not necessarily imply a sparse representation of the speech signal itself, but the two types of sparseness are related. To evaluate the sparseness of the signal representation using nonlinear ICA, we compare the output coefficients with coefficients of other transforms (MFCC, LPCC) by computing a measure of the sparseness of the feature vector itself.

One of the important measures of the sparseness of the output components is the kurtosis measure defined by

$$K(X) \;=\; E[(X-\mu_x)^4/\sigma_x^4] - 3, \qquad (44)$$

where μ_x is the mean of the random variable X, and σ_x^2 is its variance. Kurtosis is proportional to the peakiness of the probability density function of the random variable [49]. The average value of the parameter β of the generalized Gaussian probabilistic model can be used as a measure of the average kurtosis of the output components. In figure 1, the average value of the parameter β for the output coefficients that result from processing the time-domain samples of speech is shown as a function of the number of iterations of the algorithm. In figure 2, the average value of the parameter β when the speech signal is processed in the cepstral domain is shown as a function of the number of the iterations of the algorithm. The figures show that the nonlinear ICA algorithm tends to converge to output components with high kurtosis and therefore to increase the sparseness of the output coefficients. The figures also show that the nonlinear ICA algorithm with cepstral inputs tends to converge to output components with kurtosis higher than those obtained from nonlinear ICA in the time domain.

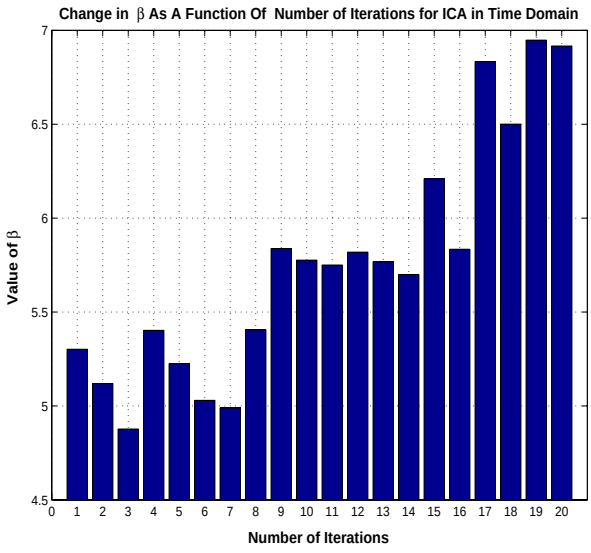

Fig. 1. The Value of β Versus Number of Iterations of Nonlinear ICA In Time Domain

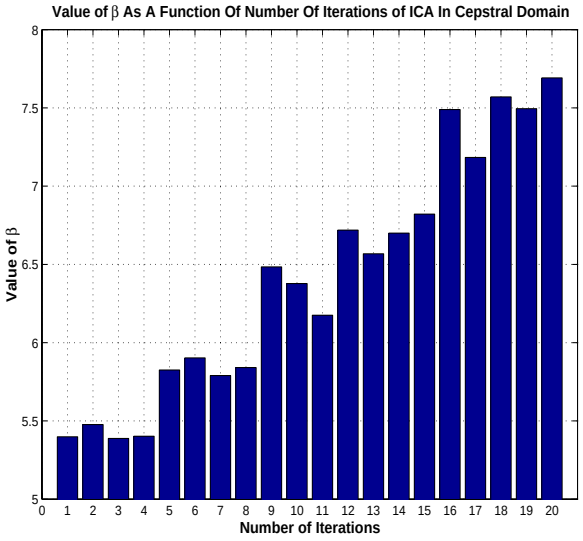

Fig. 2. The Value of β Versus Number of Iterations of Nonlinear ICA In Cepstral Domain

B. Recognition Accuracy Of Output Coefficients

There are many sources of variability in the speech signal, including linguistic information content, but also including speakers with different dialects and speaking styles, and environmental noise. Most acoustic features that have been successful in speech recognition try to model the speech signal as the convolution of the excitation signal and the vocal tract transfer function, and try to extract the vocal tract transfer function characteristics by linear predictive coding or homomorphic signal processing [50]. If linguistic and nonlinguistic information in the speech signal are independently distributed, nonlinear ICA is capable in principle of learning a mapping that approximately separates them, without the use of an explicit convolutional speech production model. In order to evaluate the success of nonlinear ICA in finding such a mapping, we performed many speech recognition experiments on the TIMIT database. The phoneme recognition accuracy achieved on TIMIT using the SUMMIT segment-based system with different features for different segments and boundaries was 75.6% and reported in [51]. The HMM speech recognizer in [52] used 12 MFCC coefficients, energy and their deltas as the acoustic features vector. It achieved a 73.7% phoneme recognition accuracy on TIMIT. While in [53], the phoneme recognition accuracy for the context-dependent models was 73.8% on TIMIT. A segment-based recognizer that was tested on TIMIT achieved a phoneme recognition accuracy of 69.5% in [54]. A speech recognizer based on recurrent networks achieved phoneme recognition accuracy of 73.4% on TIMIT in [55]. The results reported in this work like all previous results are recognition results that do not use the time alignments provided with the test data. On the other hand, phoneme classification experiments that use this time-alignment data can get classification results on TIMIT up to 81.7% as reported in [51] using heteroge-

neous measurements.

In our experiments, the 61 phonemes defined in the TIMIT database are mapped to 48 phoneme labels for each frame of speech as described in [53]. These 48 phonemes are collapsed to 39 phoneme for testing purposes as in [53]. 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 [56]. The parameters of the recognizer are trained using the training portion of the TIMIT database. The parameters of the triphone models are then tied together using the same approach as in [57].

To compare the performance of the proposed algorithm with other approaches, we generated acoustic features using LDA, linear ICA, and MLLT. We used the maximum likelihood approach to LDA [20] and kept the dimensions of the output of LDA the same as the input. We used also the maximum likelihood approach to linear ICA as described in [25] and briefly overviewed in section 2. Finally we implemented MLLT as described in [22] and briefly overviewed in section 2. 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 II. These results are obtained by using a bigram phoneme language model and by keeping the insertion error around 10% as in [53]. The table compares these recognition results to the ones obtained by MFCC, LDA, linear ICA and MLLT.

It is clear that searching for the independent components of the speech signal can not separate information in the speech signal due to linguistic variations from information due to other variations in the time domain.

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 [58]. 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 II, 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 II, 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 coefficients increases the phoneme recognition accuracy by about 7%. As shown in table II, 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 I
AN ESTIMATE OF THE DIFFERENTIAL ENTROPY OF THE FEATURES PER COEFFICIENT

Acoustic Features	Average Number of Bits
ICA in Cepstral Domain	1.52
ICA in Time Domain	1.64
MFCC	1.77
LPCC	1.85

VI. 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' coding efficiency and phoneme recognition accuracy were evaluated. The coding efficiency was found to be improved by this nonlinear mapping compared to MFCC, and LPCC coefficients. Our objective function was compared to the objective function of the sparse approximation approaches and the proximity of the two solutions was highlighted. However, the phoneme recognition accuracy based on these coefficients was clearly less than that based on MFCC. This

TABLE II
PHONEME RECOGNITION ACCURACY

Acoustic Features	Recognition Accuracy
MFCC	73.7%
Linear ICA	73.5%
LDA	73.8%
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%

means that blindly searching for the independent components of speech is not enough to be able to extract information correlated to the linguistic information contained in the speech signal, and, in case of the speech signal, independence is not the best criterion to extract meaningful components of the speech signal that are related to the actual sources of variations. This is, at least in part, because linguistic and nonlinguistic information are not entirely independent.

Second, we applied our algorithm to the MFCC features of the speech signal and its energy. Again, we compared the coding efficiency of the output coefficients to MFCC and LPCC coefficients, and the phoneme recognition accuracy of the output coefficients to LDA, linear ICA, and MLLT. In this case, the coding efficiency is improved also compared to MFCC and LPCC coefficients and even compared to non-linear ICA on time-domain data. Not only the coding efficiency but also the phoneme recognition accuracy is improved compared to MFCC, LDA, 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. We can conclude from these results that starting with well-defined features for our goal, like MFCC for phoneme recognition, our nonlinear independent component analysis can provide us with a more sparse representation that improves both the coding efficiency of the coefficients and also the recognition accuracy. The work done here supports the idea that blind information-theoretic approaches for signal analysis can not replace signal processing techniques tailored for certain application, but it can improve the performance and increase the efficiency if used to augment traditional signal processing techniques.

VII. ACKNOWLEDGMENT

This work was supported by NSF award number 0132900.

REFERENCES

[1] von Helmholtz, H., "On the Sensation of Tone as a Physiological Basis for the Study of Music," 4th. ed., A. J. Ellis, trans., Dover, New York, 1954.

[2] C. E. Shannon, "A Mathematical Theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423, July 1948.

[3] F. Atteneave, "Information Aspects Of Visual Perception," *Psychological Review*, 61, pp. 183-193, 1954.

[4] H. B. Barlow, "Sensory Mechanisms, The Reduction Of Redundancy, and Intelligence," *National Physics Laboratory Symposium No. 10, The Mechanization of Thought Processes*, 1959.

[5] H. B. Barlow, "Possible Principle Underlying The Transformation of Sensory Messages," *Sensory Communication*, W. Rosenblith ed., 1961, pp. 217-234, MIT press, Cambridge, MA.

[6] H. B. Barlow, "Unsupervised Learning," *Neural Computation*, 1, 1989, pp. 295-311. MIT press, Cambridge, MA.

[7] A. J. Bell, and T. J. Sejnowski, "An Information Maximization Approach To Blind Separation And Blind Deconvolution," *Neural Computation*, 7,1995, pp. 1129-1159.

[8] A. J. Bell, and T. J. Sejnowski, "The 'Independent Components' Of Natural Scenes Are Edge Filters," *Vision Research*, 37(23),1997, pp. 3327-3338.

[9] Toshio Irino, and Hideki Kawahara, *Signal Reconstruction from Modified Audiotry Wavelet Transform IEEE Trans. On Signal Processing*, Vol. 41, No. 12, pp. 3549-3554, December 1993.

[10] Peter F. Brown, *The acoustic-modeling problem in automatic speech recognition,* Technical report RC 12750, IBM Thomas J. Watson Research Center, 1987.

[11] Spyros Matsoukas, Thomas Colthurst, and Owen Kimball, "The 2001 Byblos English Large Vocabulary Conversational Speech Recognition System" *IEEE Proceedings of ICASSP*, pp. 721-724, Orlando, Florida, 2002.

[12] Mukund Padamanabhan, George Saon, Jing Huang, Brian Kingsbury, and Lidia Mangu, "Automatic Speech Recognition Performance on a Voicemail Transcription Task" *IEEE Trans. On Speech And Audio Processing*,Vol. 10, No. 7, pp. 433-442, October 2002.

[13] Frederick Jelinek, *Statistical Methods For Speech Recognition,* MIT Press, MIT, Cambridge, MA, 2001.

[14] *Automatic Speech and Speaker Recognition: Advanced Topics*, Chin-Hui Lee, Frank K. Soong, and Kuldip K. Paliwal editors, Kluwer Academic Publishers, 1996.

[15] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification,* Wiley, New York, NY, 2000.

[16] M. E. Tipping, and C. Bishop "Mixtures of Principal Component Analysis," *Proc. of IEEE 5th Int. Conf. Artificial Neural Networks*, 1997.

[17] S. Kajerekar, N. Malayath, and H. Hermansky, "Analysis of Sources of Variability in Speech," *Proc. of EUROSPEECH*, pp. 343-346, 1999.

[18] Hynek Hermansky, and Narendranath Malayath, "Spectral Basis Functions From Discriminant Analysis," *Proc. of Int. Conf. of Spoken Language Processing*, pp. 1379-1382, 1998.

[19] Ramalingam Hariharan, Imre Kiss, and Olli Viikki "Noise Robust Speech Parameterization Using Multiresolution Feature Extraction" *IEEE Trans. On Speech And Audio Processing*,vol. 9, No. 8, pp. 856-865, November 2001.

[20] N. Kumar, *Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition,* Ph.D. dissertation, John Hopkins Univ., Baltimore, MD, 1997.

[21] George Saon, Mukund Padmanabhan, Ramesh Gopinath, and Scott Chen, "Maximum Likelihood Discriminant Feature Spaces," *IEEE Proceedings of ICASSP*, Istanbul, Turkey, 2000.
[22] R. A. Gopinath, "Maximum Likelihood Modelling With Gaussian Distributions For Classification," *IEEE Proceedings of ICASSP*, Seattle, Washington, 1998.
[23] Aapo Hyvarinen, Juha Karhunen, and Erkki Oja, *Independent Component Analysis,* Wiley, New York, NY, 2001.
[24] Te-Won Lee, *Independent Component Analysis,* Kluwer Academic Publishers, 1998.
[25] 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.
[26] Seunggjin Choi, Heonseok Hong, Herve Glotin, Frederic Berthommier, "Multichannel signal Separation for cocktail party speech recognition: a dynamic recurrent network," *Neurocomputing*, Vol. 49, Issue 1-4, pp. 299-314, December 2002.
[27] Te-Won Lee, Anthony J. Bell, and Reinhold Orglmeister, "Blind Source Separation of Real World Signals," *IEEE Internation Conference on Neural Networks*, Houston, TX, 1997.
[28] Gil-Jin Jang, Te-Won Lee, Yung-Hwan Oh "Learning Statistically Efficient Features For Speaker Recognition," *Neurocomputing*, Vol. 49, Issue 1-4, pp. 329-348, December 2002.
[29] Jong-Hwan Lee, Ho-Young Jung, and Te-Won Lee, "Speech Feature Extraction Using Independent Component Analysis," *IEEE Proceedings of ICASSP*,Vol. 3, pp. 1631-1634, Istanbul, Turkey, 2000.
[30] I. Potamitis, N. Fakotakis, and G. Kokkinakis, "Independent Component Analysis Applied to Feature Extraction for Robust Automatic Speech Recognition," *Electronic Letters*, IEE, Vol. 36, No. 23, pp. 1977-1978, Nov. 2000.
[31] I. Potamitis, N. Fakotakis, and G. Kokkinakis, "Spectral and Cepstral projection bases constructed by Independent Component Analysis," *Proc. of Int. Conf. of Spoken Language Processing*, pp. 63-66, Beijing, China, 2000.
[32] Lawerence K. Saul, and Mazin G. Rahim "Maximum Likelihood and Minimum Classification Error Factor Analysis for Automatic Speech Recognition" *IEEE Trans. On Speech And Audio Processing*,vol. 8, No. 2, pp. 115-125, March 2000.
[33] Thomas F. Quateri, *Discrete-Time Speech Signal Processing Principles And Practice* Prientce Hall, Upper Saddle River, NJ, 2002.
[34] Thomas M. Cover, and Joy A. Thomas, *Elements of Information Theory,* Wiley, New York, NY, 1997.
[35] Athanasios Papoulis, *Probability, Random Variables, and Stochastic Processes,* McGraw-Hill, New York, 1991.
[36] Ralph Abraham, Jerrold E. Masden, *Foundation of Mechanics,* The Benjamin/Cummings Publishing Company, 1978.
[37] L. Parra, G. Deco, S. Miesbach, "Statistical independence and Novelty Detection with Information Preserving Nonlinear Maps, " *Neural Computation*, 8, pp. 260-269, 1996.
[38] K. Hornik, M. Stinchcombe, H. White, "Multilayer Feedforward Neural Networks Are Universal Approximators, " *Neural Network*, 2, pp. 359-366, 1989.
[39] A.Tikhonov, and V. Arsenin, *Solutions of Ill-Posed Problems,* Winston and Sons, Washington D. C., 1977.
[40] V. N. Vapnik, and A. Y. Chervonenkis, "Necessary and Sufficient Conditions for Consistency of The Method of Empirical Risk Minimizatition," *Pattern Recognition and Image Analysis*, 1(3), pp. 284-305, 1991.
[41] Theodoros Evgeniou, Massimiliano Pontil, Tomaso Poggio, "A Unified Framework For Regularization Networks and Support Vector Machines," *Advances in Large Margin Classifiers*, MIT Press, Cambridge, MA, 1999.
[42] Vladimir N. Vapnik, *Statistical Learning Theory,* Wiley, New York, NY, 1998.
[43] Vladimir N. Vapnik, *The Nature of Statistical Learning Theory,* Springer-Verlag, New York, 2000.
[44] A. Smola, *Learning with Kernels,* Ph.D. Thesis, Technische Universitat Berlin, 1998.
[45] Dimitri P. Bertsekas, *Nonlinear Programming,* Athena Scientific, Belmont, MA, 1999.
[46] Lawerence Rabiner, and Biing-Hwang Junag, *Fundamentals of speech recognition,* Prentice-Hall, Inc. Upper Saddle River, NJ, 1993.
[47] Todd K. Moon, "The Expectation Maximization Algorithm," *IEEE Signal Processing Magazine*, pp. 47-60, November 1996.
[48] F. Girosi, "An Equivalence Between Sparse Approximation and Support Vector Machines," *Neural Computation*, 10(6), pp. 1455-1480, 1998.
[49] James P. Leblanc, Philip L. De Leon, "Source Separation of Speech Signal Using Kurtosis Maximization," *IEEE Proceedings of ICASSP*, Seattle, Washington, 1998.
[50] John R. Deller, John H. L. Hansen, and John G. Proakis, *Discrete-Time Processing of Speech Signals,* IEEE Press, 2000.
[51] Andrew K. Halberstadt and James R. Glass, "Heterogeneous Measurements and Multiple Classifiers for Speech Recognition," *Proc. of Int. Conf. of Spoken Language Processing*, pp. 1379-1382, Sydney, Australia, 1998.
[52] S. Young, "The general use of tying in phoneme-based HMM speech recognition," *IEEE Proceedings of ICASSP*, San Francisco, CA, pp. 569-572, 1992.
[53] 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.
[54] James Glass, Jane Chang, and Michael McCandless, "A Probabilistic Framework For Feature-Based Speech Recognition," *Proc. of Int. Conf. of Spoken Language Processing*, Philadelphia, PA, pp. 2277-2280, 1996.
[55] A. Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Trans. on Neural Networks*, Vol. 5, No. 2, pp. 298-305, March 1994.
[56] S. Young, "Large Vocabulary Continous Speech Recognition," *IEEE Signal Processing Magazine*, 13(5), pp. 45-57, 1996.
[57] S. Young, and P. Woodland, "State Clustering in hidden Markov model continous speech recognition," *Computer, Speech, and Language*, 8(4), pp. 369-383, October 1994.
[58] 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.