ON-LINE MINIMUM MUTUAL INFORMATION METHOD FOR TIME-VARYING BLIND SOURCE SEPARATION

Kenneth E. Hild II, Deniz Erdogmus, and Jose C. Principe

Computational NeuroEngineering Laboratory (www.cnel.ufl.edu)
The University of Florida, Gainesville, Florida, 32611, USA
Email: k.hild@ieee.org, deniz@cnel.ufl.edu, principe@cnel.ufl.edu

ABSTRACT

The MeRMaId (Minimum Renyi's Mutual Information) algorithm for BSS (blind source separation) has previously been shown to outperform several popular algorithms in terms of data efficiency. The algorithms it compared favorably with include Hyvarinen's FastICA, Bell and Sejnowski's Infomax, and Comon's MMI (Minimum Mutual Information) methods. The drawback is that the MeRMaId algorithm has a computational complexity of O(L^2), as compared to O(L) for the other three. However, a new advancement referred to as SIG (Stochastic Information Gradient), can be used to modify the MeRMaId criterion such that the complexity is reduced to O(L). The modified criterion is then applied to the separation of instantaneously mixed sources using an on-line implementation. Simulations demonstrate that the new algorithm preserves the separation performance of the original algorithm and, in fact, compares quite favorably with several published methods.

1. INTRODUCTION

Oftentimes, it seems that some of the simplest ideas turn out to be the most useful. Take for instance the approximation that Widrow used in the steepest descent algorithm for minimizing the mean square error [1]. By using the instantaneous value of the mean square error in place of the expected value, an algorithm was developed whose use has since become ubiquitous. In fact, the algorithm, known as LMS (Least Mean Square), is nearly synonymous with gradient descent learning.

An analogous idea can also be applied to reduce the complexity of the mutual information criterion used in the MeRMaId algorithm, originally presented in [2]. In the MeRMaId algorithm, the update is found by utilizing all combinations of pairwise interactions of information particles. An "instantaneous" version of this also uses pairwise interactions, but considers only the pairwise interactions occurring between consecutive samples. The modification that allows the simplification of the MeRMaId algorithm is referred to as the Stochastic Information Gradient, or SIG, and the algorithm so modified is referred to as MeRMaId-SIG.

In the next section, the system used for instantaneous mixing and demixing is described, along with the associated notations. A brief review of the MeRMaId algorithm is then given, as well as the derivation of the SIG modification. Following this is a section which has several plots that show the separation performance results from a set of Monte Carlo simulations, and another set of plots that demonstrate the ability of the algorithm to track a time-varying mixing environment.

2. SYSTEM DESCRIPTION

The block diagram for a BSS system with N = 2 inputs and observations is given in Figure 1. Commonly, as is the case here, a pre-processor is used that spheres, or spatially whitens, the data. In this figure, the inputs are denoted as $s_i(n)$, the (whitened) observations as $x_i(n)$, and the outputs as $y_i(n)$, where $i = \{1, 2, ..., N\}$ and $n = \{1, 2, ..., L\}$. For sake of convenience, the sources will be assumed to be zero-mean.

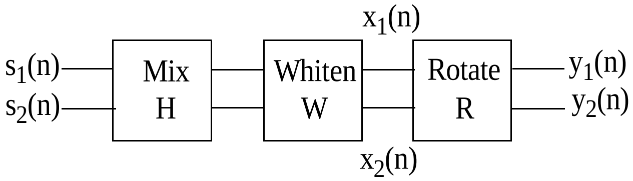

Fig. 1. Block diagram for BSS of N = 2 sources and observations.

The equations for N sources/observations for instantaneously mixed sources are given in matrix notation as, $\mathbf{x} = \mathbf{W}^T\mathbf{H}^T\mathbf{s}$ and $\mathbf{y} = \mathbf{R}^T\mathbf{x}$, where $\mathbf{s}$, $\mathbf{x}$, and $\mathbf{y}$ are the (N x L) source, sphered observation, and output matrices, respectively. In addition, $\mathbf{H}$ is the (N x N) mixing (channel) matrix, $\mathbf{W} = \Phi\Lambda^{-1/2}$ is the (N x N) sphering matrix, Φ is the matrix of eigenvectors of the autocorrelation of $\mathbf{H^Ts}$, Λ is the corresponding eigenvalue matrix, and $\mathbf{R}$ is the (N x N) demixing matrix. Notice that, due to the spatial whitening, $E[\mathbf{xx}^T] = \mathbf{I}_N$, the ($N$ x N) identity matrix.

It is known that, for instantaneous mixtures, the BSS problem can be decomposed into sphering followed by an orthogonal matrix transformation (rotation) [3]. The MeRMaId algorithm makes use of this fact; therefore, the demixing matrix, $\mathbf{R}$, is constrained to be a pure rotation. In this case, $\mathbf{R}$ is constructed from the product of $N(N\text{-}1)/2$ Givens rotations matrices [4], $\mathbf{R}_{ij}$, where $\mathbf{R}_{ij}$ equals $\mathbf{I}_N$ with elements $\mathbf{I}_N(i,i)$, $\mathbf{I}_N(i,j)$, $\mathbf{I}_N(j,i)$, and $\mathbf{I}_N(j,j)$ modified to $\cos\theta_{ij}$, $-\sin\theta_{ij}$, $\sin\theta_{ij}$, and $\cos\theta_{ij}$, respectively, and $\mathbf{I}_N(i,j)$ is the element of $\mathbf{I}_N$ located at the i^{th} row and the j^{th} column, $i = \{1, 2, ..., N\}$, and $j = \{i+1, i+2, ..., N\}$. The gradient of $\mathbf{R}$ with respect to θ_{ij} is denoted as $\nabla\mathbf{R}_{ij}$. This is equal to $\mathbf{O}_N$, an (N x N) matrix of zeros, with the elements $\mathbf{O}_N(i,i)$, $\mathbf{O}_N(i,j)$, $\mathbf{O}_N(j,i)$, and $\mathbf{O}_N(j,j)$ modified to $-\sin\theta_{ij}$, $-\cos\theta_{ij}$, $\cos\theta_{ij}$, and $-\sin\theta_{ij}$, respectively. An equivalent but more efficient formulation for the product of multiple Givens rotation matrices is provided in [5]. Notice that $\mathbf{W}$ is determined using second-order statistics, as previously explained, so the only item left to train in order to perform separation is the rotation matrix, $\mathbf{R}$. This is equivalent to finding the $N(N\text{-}1)/2$ rotation angles, θ_{ij}.

3. MeRMaId ALGORITHM

The MeRMaId algorithm is based on minimizing the Renyi's mutual information between the outputs, where Renyi's mutual information is given as [6],

$$I_{R_\alpha}(y) = \frac{1}{\alpha-1}\log\int_{-\infty}^{\infty}\left(\frac{f_Y(y)^\alpha}{\prod_{i=1}^{N} f_{Y_i}(y_i)^{\alpha-1}}\right)dy \qquad (1)$$

In the case of Shannon's entropy, the mutual information can be written as the sum of marginal entropies minus the joint entropy. This is not the case for Renyi's information. Instead, the sum of (Renyi's) marginal entropies minus the (Renyi's) joint entropy yields,

$$\sum_{i=1}^{N} H_{R_\alpha}(y_i) - H_{R_\alpha}(y) = \frac{1}{\alpha-1}\log\frac{\int_{-\infty}^{\infty} f_Y(y)^\alpha dy}{\int_{-\infty}^{\infty}\left(\prod_{i=1}^{N} f_{Y_i}(y_i)^\alpha\right)dy_i} \qquad (2)$$

Notice, however, that both (1) and (2) are minimized when and only when the joint pdf (probability density function) of y is equal to the product of the marginal pdf's of y_i, $i = \{1, 2, ..., N\}$. When this occurs, the outputs, y_i, are considered to be statistically independent. As long as the number of Gaussian distributed sources is no more than one, this is precisely the requirement for separating statistically independent sources [3]. Therefore, when the conditions for separability are met (i.e. sources are statistically independent and there is at most one Gaussian distributed source), minimizing (2) is equivalent to minimizing (1). This is the basis for the formulation of the MeRMaId algorithm. The formulation in (2) is preferred over (1) due to the existence of a non-parametric estimator for Renyi's entropy [7], along with the fact that, as discussed next, equation (2) allows a further simplification.

MeRMaId uses Parzen windows for pdf estimation, and the joint entropy in (2) involves an N-dimensional pdf. A well known result for Parzen windows is that a linear increase in the dimensionality (in this case, caused by a linear increase in the number of sources) requires an exponential increase in the number of data samples for a given accuracy, a result referred to as the "curse of dimensionality" [8]. Therefore, as the number of sources increases linearly, an algorithm based on (2) would require an exponential increase in the block size, L, to maintain a similar performance. This would cause the complexity, which is $O(L^2)$, to increase exponentially. However, this is circumvented in the MeRMaId algorithm since Renyi's entropy is invariant to rotations [6]. Hence, the joint entropy can be discarded, reducing the cost function to the sum of marginal entropies,

$$J = \sum_{i=1}^{N} H_{R_\alpha}(y_i) \qquad (3)$$

When Parzen windowing is used with a Gaussian kernel, the marginal pdf's are estimated as,

$$f_{Y_i}(y) \cong \frac{1}{L}\sum_{j=1}^{L} G(y-y_i(j)), \sigma^2) \qquad (4)$$

where G(x,σ^2) is a Gaussian pdf evaluated at x, having zero-mean and a variance of σ^2, and $y_i(j)$ is the j^{th} sample of output y_i. When this is substituted into the equation for Renyi's quadratic (alpha = 2) marginal entropy,

$$H_{R_2}(y_i) = -\log \int_{-\infty}^{\infty} f_{Y_i}(y_i)^2 dy \tag{5}$$

the following estimate for Renyi's quadratic marginal entropy is produced,

$$H_{R_2}(y_i) = -\log \frac{1}{L^2} \sum_{j=1}^{L} \sum_{k=1}^{L} G(y_i(j) - y_i(k), 2\sigma^2) \tag{6}$$

A nice feature of this derivation is that the infinite limit integral disappears without the need of any approximations or truncations [2]. Substituting (6) into (3) and taking the derivative with respect to θ_{ij} produces (7) where $(\nabla \mathbf{R}_{ij})_k$ is the k^{th} column of $\nabla \mathbf{R}_{ij}$ and $\boldsymbol{x}(m)$ is the (N x 1) vector of $\boldsymbol{x}$ at time m. The overall update equation for stochastic gradient descent is then $\Theta(n+1) = \Theta(n) - \eta\Delta\Theta(n)$, where Θ(n) and $\Delta\Theta(n)$ are (N(N-1)/2 x 1) vectors of angles and η is the step size. Due to the orthogonal constraint, the number of adaptable parameters for the demixing matrix (ignoring sphering) for MeRMaId and Comon's MMI methods, N(N-1)/2, is approximately 1/2 of that for most other algorithms, such as Infomax and FastICA, which is either N^2 or N(N-1).

4. MeRMaId-SIG

Equation (7) is the update equation for the rotation angles, which has complexity O(L^2). A straightforward method to reduce the complexity, credited to Erdogmus [9], is to use an idea analogous to the simplification of gradient descent that led to the LMS algorithm. Namely, the exact gradient expression is replaced with the "instantaneous" value of the gradient. In this case, the double summation in (7), which indexes all possible combinations of pairs of samples, is replaced with a single summation that indexes only the pairs of samples that occur consecutively. The resulting update equation, which has complexity of O(L), is given by (8), where i = {1, 2, ..., N}, j = {i+1, i+2, ..., N}, the overbar represents the angle update using SIG and the subscript of L determines the number of samples before the accumulated update is applied to the demixing matrix.

It is well known that the instantaneous gradient used in LMS converges in the mean to the actual gradient. Likewise, in [9], it is shown that the SIG update given by (8) converges in the mean to the actual information gradient given by (7), in the vicinity of the optimal solution. Therefore, there is good reason to believe that the asymptotic performance of the MeRMaId-SIG method will be similar to the original method (assuming the globally optimal solution is found).

The tap weights (or rotation angles, in this case) for MeRMaId-SIG are updated every L samples. This update is found by accumulating the individual contributions computed at each unit of time. Notice that only one memory element (as compared to L memory elements for batch methods) is needed for each input and each output of the demixer, and that the contribution to the update equation at each unit of time consists of two subtractions, four multiplications, and one function evaluation (per observation). Herein lies the advantage of using the Stochastic Information Gradient. Although SIG reduces the complexity to O(L), (perhaps) the main utility is that it allows the information theoretic criterion to be adapted in an on-line fashion.

Figure 2 shows the tap weight (angle) tracks for the case of N = 2 instantaneously mixed sources. The value of L was (somewhat arbitrarily) set to 200 and the width

$$\Delta\theta_{ij} = \frac{d}{d\theta_{ij}} \sum_{k=1}^{N} H_{R_2}(y_k) = -\sum_{k=1}^{N} \frac{\sum_{m=1}^{L} \sum_{n=1}^{L} G(y_k(m) - y_k(n), 2\sigma^2)(y_k(m) - y_k(n))(\nabla R_{ij})_k^T(\mathrm{x}(m) - \mathrm{x}(n))}{\sum_{m=1}^{L} \sum_{n=1}^{L} G(y_k(m) - y_k(n), 2\sigma^2)} \tag{7}$$

$$\Delta_L \bar{\theta}_{ij} = -\sum_{k=1}^{N} \frac{\sum_{n=1}^{L} G(y_k(n) - y_k(n-1), 2\sigma^2)(y_k(n) - y_k(n-1))(\nabla R_{ij})_k^T(\mathrm{x}(n) - \mathrm{x}(n-1))}{\sum_{n=1}^{L} G(y_k(n) - y_k(n-1), 2\sigma^2)} \tag{8}$$

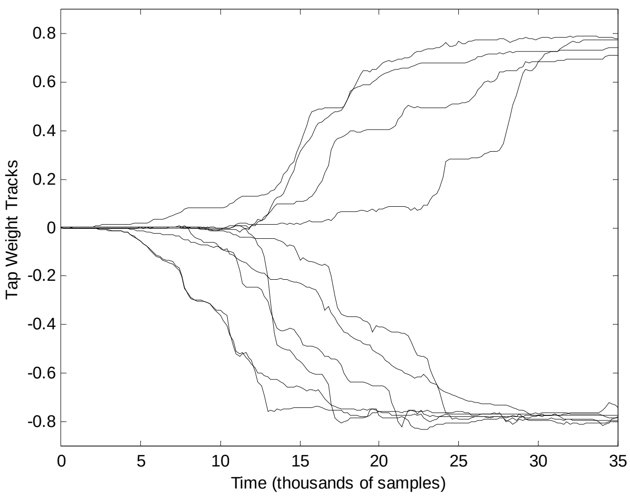

Fig. 2. Tap weight tracks for MeRMaId-SIG with N = 2 sources.

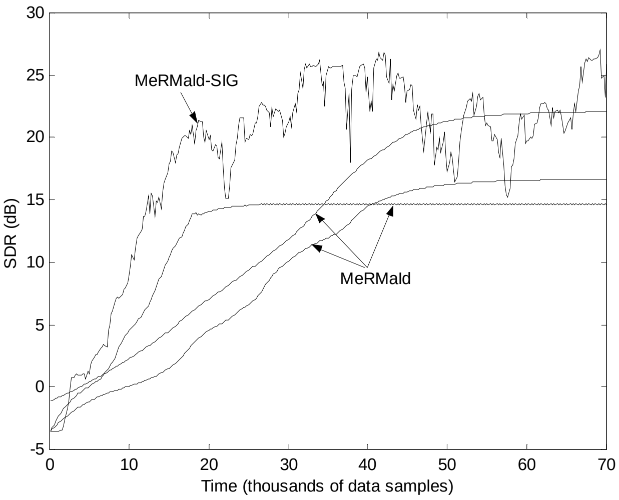

Fig. 3. SDR versus time for MeRMaId with and without SIG for N = 10 sources.

parameter for Parzen windowing, σ, was set to 0.25. Unless otherwise stated, these values are used throughout. The correct solution for this mixture is $\pi/4$ radians. For each of the ten simulations, a different pair of sources was used. Notice that the solutions all converged to either $\pi/4$ or - $\pi/4$. Both are valid solutions since a rotation of $k\pi/2$, for k any integer, from any solution is another viable solution. The difference between these multiple solutions is that the outputs are permuted and/or have a change in sign (with respect to a given solution), which are identically the indeterminacies for BSS.

An instantaneous mixture of ten sources, including speech from five male speakers and four female speakers and a segment of music, was used to compare the (standard) MeRMaId and the MeRMaId-SIG algorithms. The mixing coefficients were chosen uniformly in [-1,1] and the performance measure was chosen to be the SDR, signal-to-distortion ratio, defined as,

$$SDR = \frac{1}{N}\sum_{i=1}^{N} 10\log 10\left(\frac{max(q_i)^2}{q_i^T q_i - max(q_i)^2}\right) \qquad (9)$$

where $\mathbf{q} = \mathbf{HWR}$, $\mathbf{q}_i$ is the i^{th} column of $\mathbf{q}$, and max($\mathbf{q}_i$) is the maximum element of $\mathbf{q}_i$. This criterion effectively measures the distance of the overall mixing matrix, $\mathbf{q}$, from the product of a permutation matrix and a diagonal matrix.

Figure 3 shows the results of the comparison. Three different results are shown for MeRMaId, each using a different randomly selected 200 data samples. Notice how much smoother MeRMaId is than MeRMaId-SIG (this is still true if the step size is reduced somewhat so that both converge in approximately the same number of iterations). This can be expected when using the "instantaneous" value of the information gradient. Keep in mind that an SDR of 20 dB corresponds to nearly inaudible interference. For this comparison, the MeRMaId algorithm operates in batch mode. To wit, the 200 data samples from a given trial are used as many times as is necessary for convergence (in this example, the 200 data points are each used 70,000/200 = 350 times), whereas the MeRMaId-SIG method uses each of the 70,000 data samples exactly once.

Another comparison, shown in Figure 4, is an attempt to compare several algorithms in a manner consistent with real-time operation. In other words, each algorithm will see the data samples (in proper temporal order) only once, tap weight updates will occur every L = 1000 samples, and the memory elements (with the exception of a single memory element for MeRMaId-SIG) required for batch mode operation are not allowed. Unlike in the previous paper [2], results are not shown for Comon's MMI [3] and Hyvarinen's FastICA [11] algorithms since both are essentially batch methods. Therefore, the comparison is limited to the MeRMaId-SIG and Bell and Sejnowski's Infomax [10] methods, as well as one additional on-line, information-theoretic algorithm, Yang's MMI method [12].

All three methods utilize spatial pre-whitening. In addition, the Infomax and Yang's MMI methods use Amari's natural gradient [13], and Yang's method uses the adaptive scheme for calculating cumulants. The step sizes for all methods were chosen at or near the maximum values, such that convergence was obtained as quickly as possible. A total of 20 Monte Carlo simulations were run for each method, where the mixing matrix was selected uni-

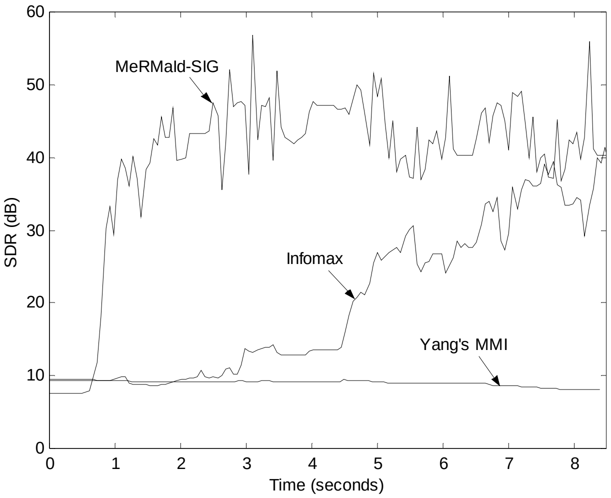

Fig. 4. SDR versus time for on-line implementations of MeRMaId-SIG, Infomax, and Yang’s MMI (*N* = 2).

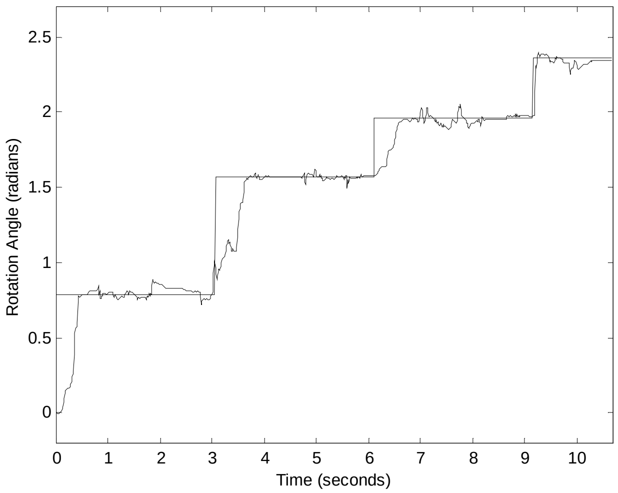

Fig. 5. Rotation angle of the mixing matrix (staircase) and the rotation angle produced by MeRMaId-SIG.

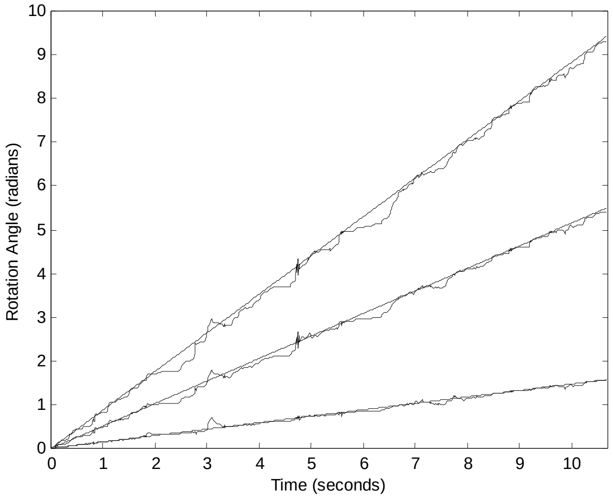

Fig. 6. Rotation angle of the mixing matrix (3 lines) and the rotation angle produced by MeRMaId-SIG.

formly in [-1,1] for each simulation. The data consists of 8.5 seconds of *N* = 2 speech sources, sampled at a rate of 16.384 kHz. Adaptation does not occur immediately because both of the sources are silent for the first 0.5 seconds. As can be seen from the ensemble-averaged plots in Figure 4, the MeRMaId-SIG method produces an SDR of 20 dB in only 0.4 real-time seconds (neglecting the first 0.5 seconds of silence), whereas Infomax took 4.2 real-time seconds, or a magnitude of order longer. Yang’s MMI method was able to reach SDR values of over 40 dB, but it consistently required much more than the 8.5 seconds of data used in this comparison (it took on the order of 30 real-time seconds to reach 20 dB SDR). In fact, the performance of Yang’s MMI method for the first 10-15 seconds of data actually dropped slightly from the initial value. Larger step sizes were tried for both the Infomax and Yang’s MMI algorithms, but they resulted in instability in the adaptation.

The comparison above was for a static environment, that is to say, the mixing matrix was held constant during the entire presentation. In a more realistic environment, the mixing matrix is a function of time (for hearing aid applications, the mixing is also convolutive, but that is not addressed here). Figures 5 and 6 show several different scenarios in which the mixing matrix was changed during the presentation of the 10.7 real-time seconds of *N* = 2 speech sources. In both cases, only the rotation angle of the mixing matrix is varied. The advantage of changing only the rotation angle is that the result is invariant to the method employed for sphering the data (a stage that is common to most BSS methods). The drawback is that BSS methods that do not constrain the demixing matrix to be a pure rotation can not be accurately tested in this fashion. For this reason, only the results for the MeRMaId-SIG algorithm are given.

Figure 5 shows the result for the case that the rotation angle of the mixing matrix is initially at 45°. The rotation angle is then changed to 90° after a little over 3 seconds have expired, 112.5° at time equal to 6 seconds, and to 135° at around time equal to 9 seconds. The staircase in this figure corresponds to the input rotation angle and the second line is the rotation angle found using the MeRMaId-SIG algorithm. Figure 6 shows 3 separate examples where the mixing matrix is initially at 0° and is then varied linearly as a function of time using 3 different velocities. As can be seen in the two figures, the MeRMaId-SIG algorithm is able to track the change in the mixing matrix quite well.

5. CONCLUSIONS

The Stochastic Information Gradient, despite (or perhaps, because of) it's simplicity, turns out to be a very useful approximation. In applications that require real-time implementations, SIG allows the use of the MeRMaId algorithm, which was previously shown to be extremely data efficient. The MeRMaId-SIG algorithm retains the data efficiency of the original algorithm and has been demonstrated to perform noticeably better than Bell and Sejnowski's Infomax and Yang's MMI (natural gradient) algorithms in a real-time application. These results indicate, at least for instantaneous mixtures, that there is hope of tracking a rapidly changing environment using a real-time, information-theoretic algorithm. Future work will consist of using a time-varying mixing matrix that has a corresponding physical correlate in terms of locations and velocities of speakers. This will be a more realistic scenario and, furthermore, will allow the comparison of the competing on-line BSS methods.

6. ACKNOWLEDGEMENTS

This work was partially supported by NSF ECF #9900394.

7. REFERENCES

[1] Bernard Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, New Jersey, 1985.

[2] Kenneth E. Hild II, Deniz Erdogmus, and Jose C. Principe, "Blind source separation using Renyi's Mutual Information," *IEEE Signal Proc. Letters*, Vol. 8, No. 6, pp. 174-176, June 2001.

[3] Pierre Comon, "Independent component analysis, a new concept?" *Signal Processing*, Vol. 36, No. 3, pp. 287-314, Apr. 1994.

[4] Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, 2nd ed., John Hopkins University Press, Baltimore, Maryland, 1989.

[5] N. Bienati, U. Spagnolini, and M. Zecca, "An adaptive blind signal separation based on the joint optimization of Givens rotations," *Proc. IEEE Intl. Conf. on Acoustic, Speech and Signal Proc., ICASSP '01*, 2001.

[6] A. Renyi, *Probability Theory*, North-Holland Publishing Company, Amsterdam, Netherlands, 1970.

[7] Jose Principe, Dongxin Xu, and John Fisher, "Information-Theoretic Learning," in *Unsupervised Adaptive Learning*, Simon Haykin Editor, John Wiley & Son, pp. 265-319, 2000.

[8] Sergios Theodoridis and Konstantinos Koutroumbas, *Pattern Recognition*, Academic Press, San Diego, California, 1999.

[9] Deniz Erdogmus and Jose C. Principe, "An on-line adaptation algorithm for adaptive system training using error entropy," submitted to *Third Intl. Workshop on Independent Component Analysis and Signal Separation, ICA '01*, 2001.

[10] Anthony J. Bell and Terrence J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, Vol. 7, No. 6, pp. 1129-1159, Nov. 1995.

[11] Aapo Hyvarinen, "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis," *IEEE Trans. Neural Networks*, Vol. 10, No. 3, pp. 626-634, 1999.

[12] Howard Hua Yang and Shun-ichi Amari, "Adaptive online learning algorithms for blind separation: maximum entropy and minimum mutual information," *Neural Computation*, Vol. 9, No. 7, pp. 1457-1482, Oct. 1997.

[13] Shun-ichi Amari, "Neural learning in structured parameter spaces - natural Riemannian gradient," *Proc. of 1996 Advances in Neural Information Proc. Systems, NIPS '96*, pp. 127-133, MIT Press, 1997.