AN EFFECTIVE DECOUPLING METHOD FOR MATRIX OPTIMIZATION AND ITS APPLICATION TO THE ICA PROBLEM

Matthew Anderson, Xi-Lin Li, Pedro Rodriguez, and Tülay Adalı

University of Maryland Baltimore County, Baltimore, MD 21250

ABSTRACT

Matrix optimization of cost functions is a common problem. Construction of methods that enable each row or column to be individually optimized, i.e., decoupled, are desirable for a number of reasons. With proper decoupling, the convergence characteristics such as local stability can be improved. Decoupling can enable density matching in applications such as independent component analysis (ICA). Lastly, efficient Newton algorithms become tractable after decoupling. The most common method for decoupling rows is to reduce the optimization space to orthogonal matrices. Such restrictions can degrade performance. We present a decoupling procedure that uses standard vector optimization procedures while still admitting nonorthogonal solutions. We utilize the decoupling procedure to develop a new decoupled ICA algorithm that uses Newton optimization enabling superior performance when the sample size is limited.

Index Terms— Independent component analysis (ICA), blind source separation (BSS), matrix optimization

1. INTRODUCTION

Optimization of cost functions with matrix parameters can occur in many domains, such as signal processing and data mining. The optimization is frequently limited to the set of full row rank matrices. Additionally, sometimes it is sufficient to consider a subset of this optimization space, namely the set of orthonormal matrices. Doing so allows the cost function to be decomposed in a manner that lets each row or column of the matrix to be optimized independently using standard vector optimization procedures. However, the subset of orthonormal matrices might be too restrictive.

In this paper, we consider an alternative optimization procedure that permits optimization over the broader class of nonorthogonal matrices. The complexity associated with matrix optimization is avoided, allowing each row to be treated independently during its optimization. Furthermore, the decoupling enables the development of Newton optimization algorithms.

This work is supported by the NSF grants NSF-III 1017718 and NSF-CIF 1117056.

We begin by providing some background on the maximum likelihood approach to ICA, as it provides a primary example of the matrix optimization problem of interest. In Section 3, the method for decoupling each row of a nonorthogonal matrix optimization parameter is described. Then, we apply this decoupling approach to introduce the decoupled ICA (D-ICA) algorithm. Lastly, we compare using simulations the performance of D-ICA with several existing ICA algorithms.

2. BACKGROUND AND MOTIVATION FOR DECOUPLED LEARNING

In a number of problems, one has to perform optimization with respect to a matrix parameter using iterative optimization techniques. This is especially the case in approaches in latent variable analysis such as ICA and nonnegative matrix factorization (NMF). Without loss of generality, the development here is based on maximum likelihood ICA. Extensions to other problems, especially to NMF and other blind source separation (BSS) problems is quite straightforward.

In BSS problems an N dimensional observation vector, $\mathbf{x}(t)$ is observed T times. The sources can be blindly identified upto a scaling and permutation ambiguity via ICA when a noiseless instantaneous linear mixing model of independent sources is valid. Throughout we assume that the samples of $\mathbf{x}(t)$ are independently and identically distributed (i.i.d.) and the sample index, t, is henceforth suppressed. Specifically, ICA requires $\mathbf{x} = \mathbf{A}\mathbf{s}$, where the square, $N \times N$ mixing matrix, $\mathbf{A}$, must be invertible and the source vector components are statistically independent so that the joint probability distribution function (pdf) of the source vector $\mathbf{s}$ can be factored as a product of marginal pdfs, i.e., $p(\mathbf{s}) = p(s_1) \cdot \ldots \cdot p(s_N)$. In order for all the sources to be identified via ICA, at most one source can be normally distributed. In ICA, the estimates of the source vector are obtained by $\mathbf{y} = \mathbf{W}\mathbf{x}$, where $\mathbf{W}$ is an estimate of the "demixing" matrix.

One principled ICA approach is to minimize the mutual information of the estimated source components as expressed in the following cost function,

$$\mathcal{J}_{\text{ICA}_1} = \sum_{n=1}^{N} \mathcal{H}[y_n] - \log|\det(\mathbf{W})| - C_1, \qquad (1)$$

where $\mathcal{H}[y_n]$ is the (differential) entropy of the nth estimated source, and the entropy of the observations $\mathcal{H}[\mathbf{x}]$ is a constant with respect to (w.r.t.) $\mathbf{W}$, denoted by C_1.

There are several interesting properties of the ICA cost function given in (1). First, note that it is equivalent to maximum likelihood estimation [1]. The role of the $\log|\det(\mathbf{W})|$ portion of the cost function is to act as a regularization term. Since entropy is not scale invariant, i.e., $\mathcal{H}[z] \neq \mathcal{H}[az]$ for $a \neq 1$, then without the regularization term the cost function could be minimized by scaling the estimated sources. A simplification of the cost function can be had by restricting the set of permissible demixing matrices to those that achieve the following *necessary condition* for estimating independent sources, namely requiring that the estimated sources are second-order uncorrelated, or $\mathcal{E}\left[\mathbf{y}\mathbf{y}^T\right] = \mathbf{D}$, where $\mathbf{D}$ is a full rank diagonal matrix. This necessary condition strictly holds as $T \to \infty$, and is achieved by a whitening matrix, $\mathbf{U}$, such that $\mathcal{E}\left[\mathbf{z}\mathbf{z}^T\right] = \mathbf{I}$, where $\mathbf{z} = \mathbf{U}\mathbf{x}$. The set of all demixing matrices which meet this second-order requirement can then be expressed as $\mathbf{W} = \mathbf{V}\mathbf{U}$, where $\mathbf{V}$ is an orthonormal matrix to be estimated via optimization of this simplified ICA cost function,

$$\mathcal{J}_{\text{ICA}_2} = \sum_{n=1}^{N} \mathcal{H}[y_n] - C_2 - C_1. \tag{2}$$

We have used $C_2 \triangleq \log|\det(\mathbf{W})|$, since this is now a constant w.r.t. the orthogonal optimization parameter $\mathbf{V}$. It should be clear that prewhitening data using $\mathbf{U}$ does not make (1) and (2) equivalent. Data can be whitened prior to using (1) but the data *must* be whitened to use (2). In the above and throughout the remainder of the paper, we restrict our discussion to the real domain and note that similar results can be achieved with proper analysis for the complex domain.

The more restrictive cost function given in (2) is widely used as it allows the estimation of each source component using standard vector optimization procedures. If T is not sufficiently large then the accuracy of the whitening matrix, $\mathbf{U}$, is degraded and the restriction on decomposing $\mathbf{W}$ as the product of an orthogonal matrix and a whitening matrix may degrade source estimation performance. This is the primary motivation for considering the more general cost function of (1) rather than the simpler cost function of (2). For simplicity, we will refer to (1) and (2) as the nonorthogonal and orthogonal ICA cost functions, respectively. Similarly, orthogonal/nonorthogonal cost functions exist in other optimization problems; ICA only serves as one convenient example.

3. DECOUPLING PROCEDURE

A useful decoupling for some matrix optimization problems is first introduced in [2] for designing an approximate joint diagonalization algorithm. Variants of this decoupling procedure have been subsequently used in [3, 4] for designing ICA algorithms and in [5] for designing joint blind source separation (JBSS) algorithms. Here we present a distinct derivation of this decoupling "trick" using basic linear algebra. The decoupling procedure can be applied to cost functions that are optimized over the set of full row-rank matrices and have a regularization term based on $\sqrt{\det(\mathbf{W}\mathbf{W}^T)}$. For full-rank matrices this is equivalent to $|\det(\mathbf{W})|$.

Let the matrix to be estimated be expressed in terms of vectors, $\mathbf{W} = [\mathbf{w}_1 \dots \mathbf{w}_M]^T \in \mathbb{R}^{M\times N}$, where $M \leq N$. We wish to decouple the estimation of each row in $\mathbf{W}$, $\mathbf{w}_m^T$, $1 \leq m \leq M$. To do so, we will denote the other $M-1$ rows in $\mathbf{W}$ as $\tilde{\mathbf{W}}_m = [\mathbf{w}_1 \dots \mathbf{w}_{m-1}\mathbf{w}_{m+1} \dots \mathbf{w}_M]^T \in \mathbb{R}^{(M-1)\times N}$. By using a permutation matrix, $\mathbf{P}_{m,M}$, we can exchange the mth and Mth rows of $\mathbf{W}$ using $\mathbf{P}_{m,M}\mathbf{W}$. This enables us to use the determinant of partitioned matrices given in [6] to write

$$\begin{aligned}
\det\left(\mathbf{W}\mathbf{W}^T\right) &= \det\left(\mathbf{P}_{m,M}\mathbf{W}\mathbf{W}^T\mathbf{P}_{m,M}^T\right) \\
&= \det\left(\begin{bmatrix} \tilde{\mathbf{W}}_m \\ \mathbf{w}_m^T \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{W}}_m^T & \mathbf{w}_m \end{bmatrix}\right) \\
&= \det\left(\tilde{\mathbf{W}}_m\tilde{\mathbf{W}}_m^T\right)\mathbf{w}_m^T\tilde{\mathbf{H}}_m\mathbf{w}_m,
\end{aligned} \tag{3}$$

where $\tilde{\mathbf{H}}_m \triangleq \mathbf{I} - \tilde{\mathbf{W}}_m^T\left(\tilde{\mathbf{W}}_m\tilde{\mathbf{W}}_m^T\right)^{-1}\tilde{\mathbf{W}}_m \in \mathbb{R}^{N\times N}$. Note that $\mathbf{w}_m^T\tilde{\mathbf{H}}_m\mathbf{w}_m$ is the Schur complement of $\tilde{\mathbf{W}}_m\tilde{\mathbf{W}}_m^T$ in $\mathbf{W}\mathbf{W}^T$.

Recall from linear algebra that the least squares solution to $\mathbf{A}\mathbf{x} = \mathbf{b} \in \mathbb{R}^q$, where $\mathbf{A} \in \mathbb{R}^{q\times r}$ has full column rank and $r \leq q$, can be solved using the normal system of equations, $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$, or $\mathbf{x} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$. Thus the projection vector, $\mathbf{p} \triangleq \mathbf{A}\mathbf{x} = \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}$, makes it clear that $\mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$ is a projection matrix that maps vectors onto the column space of $\mathbf{A}$. Orthonormal projection matrices possess many useful properties, one of which is the orthogonal complement, i.e., the null space of $\mathbf{A}$ is given by $\mathbf{I} - \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$.

With the recollection above, it is clear that $\tilde{\mathbf{H}}_m$ is an orthogonal complement projection matrix for the space spanned by the rows of $\tilde{\mathbf{W}}_m$. Due to space considerations, we now consider the most common case when $M = N$, i.e., $\mathbf{W}$ is invertible, then by the chosen decomposition of $\mathbf{W}$, we have that $\tilde{\mathbf{H}}_m$ is a rank one matrix. More explicitly, $\tilde{\mathbf{H}}_m = \mathbf{h}_m\mathbf{h}_m^T$, where $\mathbf{h}_m \in \mathbb{R}^{N\times 1}$ and $\|\mathbf{h}_m\| = 1$ due to the requirement that orthonormal projections matrices have eigenvalues of 1 or 0 only. To clarify, $\mathbf{h}_m$ is any vector such that $\tilde{\mathbf{W}}_m\mathbf{h}_m = \mathbf{0} \in \mathbb{R}^{(N-1)\times 1}$.

Expressing the desired quantity, $\sqrt{\det(\mathbf{W}\mathbf{W}^T)}$, when $M = N$, in terms of the result above in (3) we have

$$\sqrt{\det(\mathbf{W}\mathbf{W}^T)} = \sqrt{\det\left(\tilde{\mathbf{W}}_m\tilde{\mathbf{W}}_m^T\right)}\,|\mathbf{w}_m^T\mathbf{h}_m|. \tag{4}$$

A geometric interpretation of (4) is to consider the left hand side as a volumetric term so that the first term on the right

hand side is an 'area' term associated with the submatrix $\tilde{\mathbf{W}}_m$ and the last term gives a 'height' measure.

Using the result above one can readily compute derivatives of $\sqrt{\det(\mathbf{W}\mathbf{W}^T)}$ w.r.t. each row $\mathbf{w}_m$. It is actually more convenient for our purposes to compute the derivative of $\log\sqrt{\det(\mathbf{W}\mathbf{W}^T)}$,

$$\frac{\partial \log\sqrt{\det(\mathbf{W}\mathbf{W}^T)}}{\partial \mathbf{w}_m} = \frac{\mathbf{h}_m}{\mathbf{w}_m^T\mathbf{h}_m}, \tag{5}$$

and the associated Hessian is

$$\frac{\partial^2 \log\sqrt{\det(\mathbf{W}\mathbf{W}^T)}}{\partial \mathbf{w}_m \partial \mathbf{w}_m^T} = \frac{-1}{(\mathbf{w}_m^T\mathbf{h}_m)^2}\tilde{\mathbf{H}}_m. \tag{6}$$

The difference between the original derivation and use of [2] with the decoupling presented here is that the former used a 2×2 block matrix decomposition of $\mathbf{W}$ rather than the essentially 2×1 block matrix decomposition used here. Additionally, with the presented decoupling derivation, the origin of the vector $\mathbf{h}_n$ is clear and consistent with the decoupling used in [3, 4, 5].

4. DECOUPLED MAXIMUM LIKELIHOOD ICA

In this section, we use the decoupling procedure derived in Section 3 to develop a new decoupled maximum likelihood ICA algorithm. To do so, we use the more general ICA cost function given in (1), whose minimization corresponds to minimizing the mutual information of the estimated components. The demixing matrix estimate can be optimized by minimizing the cost function using either gradient or Newton-based optimization methods. Previous optimization approaches compute the derivative of (1) w.r.t. the entire demixing matrix. To achieve faster convergence and to avoid computing the inverse of the demixing matrix the natural (relative) gradient algorithm can be used [7]. Although Newton optimization procedures for updating the demixing matrix are theoretically possible, they are generally not practical for moderate to large values of N because the dimension of the Hessian grows as N^2. Here, we achieve a Newton-based algorithm by using the decoupling procedure.

The gradient of the ICA cost function (1) w.r.t. the nth row of $\mathbf{W}$ is

$$\frac{\partial \mathcal{J}_{\text{ICA}_1}}{\partial \mathbf{w}_n} = \mathcal{E}\left[\phi(y_n)\,\mathbf{x}\right] - \frac{\mathbf{h}_n}{\mathbf{w}_n^T\mathbf{h}_n}, \tag{7}$$

where we have applied the chain rule and used $\phi(y_n)$ to denote the scalar quantity $-d\log p(y_n)/dy_n$, sometimes termed the score function. For a Newton update, the Hessian can be computed using

$$\frac{\partial^2 \mathcal{J}_{\text{ICA}_1}}{\partial \mathbf{w}_n \partial \mathbf{w}_n^T} = \mathcal{E}\left[\phi'(y_n)\,\mathbf{x}\mathbf{x}^T\right] + \frac{1}{(\mathbf{w}_n^T\mathbf{h}_n)^2}\tilde{\mathbf{H}}_n, \tag{8}$$

where $\phi'(y_n) = d\phi(y_n)/dy_n$ is defined provided pdf is twice differentiable. Given such a score function, a Newton update of the nth demixing vector is

$$\mathbf{w}_{n,\text{new}} \leftarrow \mathbf{w}_{n,\text{old}} - \mu\left(\frac{\partial^2 \mathcal{J}_{\text{ICA}_1}}{\partial \mathbf{w}_n \partial \mathbf{w}_n^T}\right)^{-1}\frac{\partial \mathcal{J}_{\text{ICA}_1}}{\partial \mathbf{w}_n}, \tag{9}$$

where $\mu > 0$ is a step-size parameter, which is one for a truly Newton algorithm but its value can be adjusted to control the convergence speed.

The computations required to compute the Hessian directly as expressed in (8) and the Hessian inverse as required in (9) are potentially computationally burdensome. To reduce the computational operations required to iteratively update $\mathbf{w}_n$ we consider prewhitened data. Then a simplifying assumption, $\mathcal{E}\left[\phi'(y_n)\,\mathbf{x}\mathbf{x}^T\right] \approx \mathcal{E}\left[\phi'(y_n)\right]\mathbf{I}$, used in [8] to derive the FastICA algorithm can be considered. Using this simplification provides the following Hessian approximation,

$$\frac{\partial^2 \mathcal{J}_{\text{ICA}_1}}{\partial \mathbf{w}_n \partial \mathbf{w}_n^T} \approx \mathcal{E}\left[\phi'(y_n)\right]\mathbf{I} + \frac{1}{(\mathbf{w}_n^T\mathbf{h}_n)^2}\tilde{\mathbf{H}}_n. \tag{10}$$

Furthermore, by the matrix inversion lemma [6], we have the following expression for the Hessian inverse,

$$\begin{aligned}\left(\frac{\partial^2 \mathcal{J}_{\text{ICA}_1}}{\partial \mathbf{w}_n \partial \mathbf{w}_n^T}\right)^{-1} &\approx \left(\mathcal{E}\left[\phi'(y_n)\right]\mathbf{I} + \frac{1}{(\mathbf{w}_n^T\mathbf{h}_n)^2}\tilde{\mathbf{H}}_n\right)^{-1} \\ &= \gamma_n\mathbf{I} - \frac{\gamma_n^2}{\alpha_n+\gamma_n}\tilde{\mathbf{H}}_n,\end{aligned} \tag{11}$$

where $\gamma_n^{-1} \triangleq \mathcal{E}\left[\phi'(y_n)\right]$ and $\alpha_n \triangleq \left(\mathbf{w}_n^T\mathbf{h}_n\right)^2$. Using the approximation for the Hessian inverse of (11) in (9), we have the following computationally efficient quasi-Newton update rule for D-ICA

$$\mathbf{w}_{n,\text{new}} \leftarrow \mathbf{w}_{n,\text{old}} - \mu\left(\gamma_n\mathbf{I} - \frac{\gamma_n^2}{\alpha_n+\gamma_n}\tilde{\mathbf{H}}_n\right)\frac{\partial \mathcal{J}_{\text{ICA}_1}}{\partial \mathbf{w}_n}. \tag{12}$$

4.1. Algorithm Details

In the previous subsection, the two fundamental update equations for D-ICA are given in (9) and (12). In this subsection, we provide implementation details.

A prudent practice is to initialize nonorthogonal algorithms with solutions from (faster) orthogonal algorithms. For this paper, we have chosen to use the solution of the popular orthogonal FastICA [8] to initialize the estimate of the demixing matrix. By doing so, the nonorthogonal D-ICA algorithm refines the orthogonal FastICA solution so that only a few iterations by D-ICA are necessary to converge. Thus concerns about the additional computational cost of the decoupling procedure can be reduced considerably. Additionally, a fast recursive method for computing $\mathbf{h}_n$ can be used [4].

In each iteration, all the rows of the demixing matrix are updated using (12). After each update the demixing row vectors are normalized to have unit length. As the algorithm converges measures of changes in the demixing vector estimates between iterations, such as $\theta_n \triangleq 1 - |\mathbf{w}_{n,\text{old}}^T \mathbf{w}_{n,\text{new}}|$, become very small. We deem that convergence is achieved when $\max(\theta_1, \ldots, \theta_N) < \epsilon$, where ϵ is a small positive number with a typical value of 10^{-6}. The quasi-Newton update of (12) is used until the last iteration, which uses the more computationally expensive Newton update of (9). For both the exact and quasi-Newton update rules we have used $\mu = 1$ for the step-size parameter.

Lastly, to implement D-ICA we specify an a priori distribution, namely a member of the inverse-cosh family of distributions, $p(y) \propto 1/\cosh^{1/\beta}(\beta y)$. In particular, we let $\beta = 1/2$, then $\phi(y) = \tanh(y/2)$ and $\phi'(y) = \left(1 - \phi^2(y)\right)/2$.

4.2. Algorithm Performance

To demonstrate the performance, we simulate sources as i.i.d. samples of the inverse-cosh distribution described above. For comparison, we compare the performance of D-ICA with FastICA and Infomax [9]. Both Infomax and D-ICA use the symmetric FastICA solution for initialization. All three algorithms are using the same source density matching tangent hyperbolic score function. To compare performance we consider the normalized inter-symbol-interference (ISI) metric [10, 11]:

$$\text{ISI}(\mathbf{G}) \triangleq \frac{1}{2N(N-1)} \left[\sum_{n=1}^{N} \left(\sum_{m=1}^{N} \frac{|g_{n,m}|}{\max_p |g_{n,p}|} - 1 \right) + \sum_{m=1}^{N} \left(\sum_{n=1}^{N} \frac{|g_{n,m}|}{\max_p |g_{p,m}|} - 1 \right) \right],$$

where $\mathbf{G} \triangleq \mathbf{WA}$ and $g_{m,n}$ is the mth and nth element of $\mathbf{G}$.

In each experiment, the elements of the mixing matrix are drawn from the standard normal distribution. The average normalized ISI of 50 trials for various number of sources and sample sizes are shown in Fig. 1. For this example, the D-ICA algorithm provides equal or better performance than FastICA and Infomax for all experimental settings and provides the largest benefit when the sample size is small.

5. CONCLUSIONS

The ability to decouple the rows of nonorthogonal matrix optimization parameters can be preferred to the more restrictive orthogonal decoupling. The benefits of decoupling can be exhibited in terms of simplified algorithm design and improved optimization performance as demonstrated here. Additionally, decoupling enables density matching for ICA algorithms.

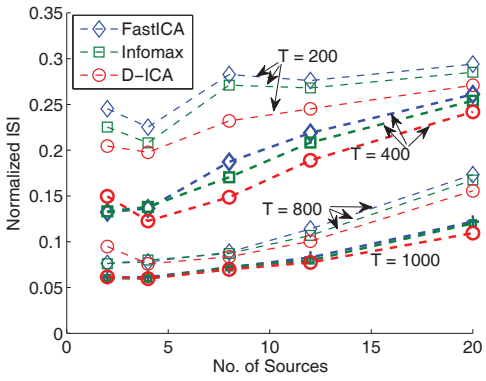

Fig. 1. Average normalized ISI of 50 trials for 200, 400, 800, and 1000 samples versus number of sources.

6. REFERENCES

[1] J.-F. Cardoso, "Infomax and maximum likelihood for blind source separation," *Signal Processing Letters, IEEE*, vol. 4, no. 4, pp. 112–114, Apr. 1997.

[2] X.-L. Li and X.-D. Zhang, "Nonorthogonal joint diagonalization free of degenerate solution," *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1803–1814, May 2007.

[3] X.-L. Li and T. Adalı, "Complex independent component analysis by entropy bound minimization," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 57, no. 7, pp. 1417–1430, Jul. 2010.

[4] ——, "Independent component analysis by entropy bound minimization," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5151–5164, Oct. 2010.

[5] M. Anderson, T. Adalı, and X.-L. Li, "Joint blind source separation of multivariate Gaussian sources: Algorithms and performance analysis," *IEEE Trans. Signal Process.*, 2012, to be published.

[6] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge: Cambridge University Press, 1985.

[7] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 1st ed. Academic Press, 2010.

[8] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *Neural Networks, IEEE Transactions on*, vol. 10, no. 3, pp. 626–634, May 1999.

[9] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[10] E. Moreau and O. Macchi, "A one stage self-adaptive algorithm for source separation," *Acoustics, Speech, and Signal Processing*, vol. 3, pp. 49–52, 1994.

[11] S. Choi, A. Cichocki, L. Zhang, and S. Amari, "Approximate maximum likelihood source separation using the natural gradient," in *Wireless Communications, 2001. (SPAWC '01). 2001 IEEE Third Workshop on Signal Processing Advances in*, 2001, pp. 235–238.