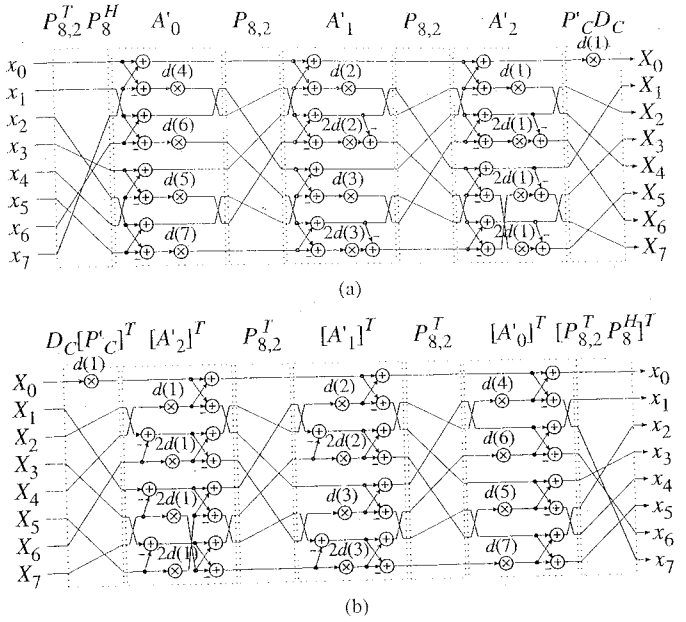

Fig. 4. Signal flow graphs of constant geometry algorithm for (a) forward and (b) inverse 8-point DCT-II.

algorithms where the number of multiplications reaches the theoretical lower bound, e.g., in algorithm reported by Loeffler *et al.* [9], the regular structure of the proposed algorithm provides advantages in implementations. The constant topology interconnections between the processing stages can be realized with simple register-based interconnection networks as described, e.g., in [7]. This allows the constant geometry algorithm to be mapped onto varying number of arithmetic resources; thus, the degree of parallelism in the final architecture can be tailored to given throughput requirements, as described in [1] and [5]. The proposed algorithm lends itself not only to parallel hardware implementations but to software implementations on processors containing parallel computational resources as well. The known systematic mappings to parameterized architectures from [1] can also be applied for deriving algorithms with similar property for other discrete trigonometric transforms.

Acknowledgment

The authors would like to thank the anonymous referees for the constructive comments and suggestions.

References

[1] J. Astola and D. Akopian, "Architecture-oriented regular algorithms for discrete sine and cosine transforms," *IEEE Trans. Signal Processing*, vol. 47, pp. 1109–1124, Apr. 1999.

[2] S. Winograd, "On computing the DFT," *Math. Comput.*, vol. 32, no. 1, pp. 175–199, Jan. 1978.

[3] P. Duhamel and M. Vetterli, "Fast Fourier transforms: A tutorial and a state of the art," *Signal Process.*, vol. 19, no. 4, pp. 259–299, Apr. 1990.

[4] J. Cooley and J. Tukey, "An algorithm for the machine calculation of the complex Fourier series," *Math. Comput.*, vol. 19, pp. 297–301, Apr. 1965.

[5] F. Argüello, J. Bruguera, R. Doallo, and E. L. Zapata, "Parallel architecture for fast transforms with trigonometric kernel," *IEEE Trans. Parallel Distrib. Syst.*, vol. 5, pp. 1091–1099, Oct. 1994.

[6] M. Davio, "Kronecker products and shuffle algebra," *IEEE Trans. Comput.*, vol. 30, pp. 116–125, Feb. 1981.

[7] D. Akopian, J. Takala, J. Astola, and J. Saarinen, "Multistage interconnection networks for k/n rate Viterbi decoders," in *Proc. IEEE Global Telecommun. Conf.*, Sydney, Australia, Nov. 8–12, 1998, pp. 845–850.

[8] Z. Wang, "Pruning the fast discrete cosine transform," *IEEE Trans. Commun.*, vol. 39, pp. 640–643, May 1991.

[9] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Glasgow, U.K., May 23–26, 1989, pp. 988–991.

On Gradient Adaptation with Unit-Norm Constraints

Scott C. Douglas, Shun-ichi Amari, and S.-Y. Kung

***Abstract*—In this correspondence, we describe gradient-based adaptive algorithms within parameter spaces that are specified by $\|\mathbf{w}\| = 1$, where $\|\cdot\|$ is any vector norm. We provide several algorithm forms and relate them to true gradient procedures via their geometric structures. We also give algorithms that mitigate an inherent numerical instability for L_2-norm-constrained optimization tasks. Simulations showing the performance of the techniques for independent component analysis are provided.**

I. Introduction

Consider the following: Given a cost function $\mathcal{J}(\mathbf{w})$ for the parameter vector $\mathbf{w} = [w_1 \;\; w_2 \;\; \cdots \;\; w_n]^T$

$$\text{maximize} \quad \mathcal{J}(\mathbf{w}) \tag{1}$$

$$\text{such that} \quad \|\mathbf{w}\| = C \tag{2}$$

where $\|\mathbf{w}\|$ is any vector norm, and C is a positive constant. This problem forms the basis for many useful tasks in communications, control, numerical analysis, signal processing, and statistics [1]–[18]. Note that (2) imposes a geometric structure to the parameter space. Consider

$$\|\mathbf{w}\|_p = \left(\sum_{i=1}^{n} |w_i|^p \right)^{1/p} \tag{3}$$

to be the L_p norm of $\mathbf{w}$, where $1 \leq p \leq \infty$. When $p = 2$, the parameter vectors satisfying (2) form an n-dimensional hypersphere of radius C. When $p = 1$, (2) defines an n-dimensional hyperpolyhedron with vertices $C\mathbf{e}_i = [0 \;\; \cdots \;\; 0 \;\; C \;\; 0 \;\; \cdots \;\; 0]^T$. When $p = \infty$, (2) defines an n-dimensional hypercube.

This correspondence considers gradient-based iterative algorithms for solving (1) and (2). To our knowledge, a general comparison of such approaches has not been provided in the signal processing literature. In the case of the L_2-norm parameter constraint, we also investigate the numerical issues surrounding these gradient methods and describe *self-stabilized* methods that implicitly maintain (2) without periodic renormalization, additional penalty terms, and costly divides or

Manuscript received August 10, 1998; revised November 12, 1999. This work was supported in part by the Office of Research and Development under Contract 98F135700-000. The associate editor coordinating the review of this paper and approving it for publication was Dr. Ali H. Sayed.

S. C. Douglas is with the Department of Electrical Engineering, School of Engineering and Applied Science, Southern Methodist University, Dallas, TX 75275 USA (e-mail: douglas@seas.smu.edu).

S. Amari is with the Laboratory for Information Synthesis, RIKEN Brain Science Institute, Saitama, Japan.

S.-Y. Kung is with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA.

Publisher Item Identifier S 1053-587X(00)04072-1.

square roots. An application to minimum-kurtosis independent component analysis (ICA) is provided.

II. General Forms of Gradient Algorithms

Gradient algorithms for (1) and (2) have different forms, depending on how the constraint is imposed. We give general algorithm forms in this section. Without loss of generality, set $C = 1$, and let

$$\begin{aligned} \mathbf{g}(k) &= \mu(k)\frac{\partial \mathcal{J}(\mathbf{w}(k))}{\partial \mathbf{w}} \\ &= \mu(k)\left[\frac{\partial \mathcal{J}(\mathbf{w})}{\partial w_1} \cdots \frac{\partial \mathcal{J}(\mathbf{w})}{\partial w_n}\right]^T\Bigg|_{\mathbf{w}=\mathbf{w}(k)} \end{aligned} \tag{4}$$

be the scaled gradient of $\mathcal{J}(\mathbf{w})$ in Euclidean space evaluated at $\mathbf{w} = \mathbf{w}(k)$, where $\mu(k)$ is a positive step-size parameter. We also define

$$\begin{aligned} \mathbf{h}_{\mathbf{g}}(k) &= \left(\mathbf{I} - \frac{\mathbf{v}(k)\mathbf{v}^T(k)}{\|\mathbf{v}(k)\|_2^2}\right)\mathbf{g}(k) \quad \text{and} \\ \mathbf{v}(k) &= \frac{\partial \|\mathbf{w}\|}{\partial \mathbf{w}}\Bigg|_{\mathbf{w}=\mathbf{w}(k)} \end{aligned} \tag{5}$$

where $\|\mathbf{g}(k)\|_2 < \infty$. Geometrically, $\mathbf{h}_{\mathbf{g}}(k)$ is tangent to the surface $\|\mathbf{w}\| = 1$ at $\mathbf{w} = \mathbf{w}(k)$ and is called the *tangent gradient* of $\mathcal{J}(\mathbf{w})$ in the constraint space. In differential geometry, the set of all such vectors is known as the *tangent space* of the surface $\|\mathbf{w}\| = 1$ at $\mathbf{w}(k)$ [17].

Lagrange Multiplier Method [1]. Define the augmented cost function

$$\hat{\mathcal{J}}(\mathbf{w}) = \mathcal{J}(\mathbf{w}) + \lambda\|\mathbf{w}\| \tag{6}$$

where the Lagrange multiplier λ is chosen to satisfy $\|\mathbf{w}\| = 1$. Then, one update for $\mathbf{w}(k)$ is

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \mu(k)\frac{\partial \hat{\mathcal{J}}(\mathbf{w}(k))}{\partial \mathbf{w}} \\ &= \mathbf{w}(k) + \mathbf{g}(k) + \lambda(k)\mathbf{v}(k) \end{aligned} \tag{7}$$

where $\lambda(k) = \lambda\mu(k)$. In this case, the sequence $\lambda(k)$ should satisfy $\lim_{k\to\infty}\|\mathbf{w}(k)\| = 1$. If $\|\mathbf{w}(k)\| = 1$ is imposed at each k and if such a solution exists, $\lambda(k)$ satisfies

$$\|\mathbf{w}(k) + \mathbf{g}(k) + \lambda(k)\mathbf{v}(k)\| = 1. \tag{8}$$

Consider the case where $\|\mathbf{w}\| = \|\mathbf{w}\|_2$ is the L_2 norm. Then, $\mathbf{v}(k) = \mathbf{w}(k)$ in (5) and (7) is

$$\mathbf{w}(k+1) = \alpha(k)\mathbf{w}(k) + \mathbf{g}(k) \tag{9}$$

where $\alpha(k) = 1 + \lambda(k)$ for convenience. The value of $\alpha(k)$ satisfying $\|\mathbf{w}(k+1)\|_2 = 1$ is

$$\alpha(k) = \sqrt{1 - \|\mathbf{g}(k)\|_2^2 + [\mathbf{w}^T(k)\mathbf{g}(k)]^2} - \mathbf{w}^T(k)\mathbf{g}(k). \tag{10}$$

In this algorithm, $\mathbf{w}(k)$ is rotated to $\mathbf{w}(k+1)$ in the direction of $\mathbf{h}_{\mathbf{g}}(k)$ by an angle $\theta(k)$, where the form of $\theta(k)$ is given in the first entry of Table I.

Coefficient Normalization Method [3], [4]. The well-known gradient update

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{g}(k) \tag{11}$$

TABLE I
Angles of Rotation for L_2-Norm Constrained Adaptation

Algorithm	*Angle of Rotation*
Lagrange	$\sin(\theta(k)) = \sqrt{\|\mathbf{g}(k)\|_2^2 - [\mathbf{w}^T(k)\mathbf{g}(k)]^2}$
Normalized	$\sin(\theta(k)) = \sqrt{\dfrac{\|\mathbf{g}(k)\|_2^2 - (\mathbf{w}^T(k)\mathbf{g}(k))^2}{1 + 2\mathbf{w}^T(k)\mathbf{g}(k) + \|\mathbf{g}(k)\|_2^2}}$
Tangent	$\tan(\theta(k)) = \sqrt{\|\mathbf{g}(k)\|_2^2 - [\mathbf{w}^T(k)\mathbf{g}(k)]^2}$
True Gradient	$\theta(k) = \sqrt{\|\mathbf{g}(k)\|_2^2 - [\mathbf{w}^T(k)\mathbf{g}(k)]^2}$

performs unconstrained maximization of $\mathcal{J}(\mathbf{w})$. The coefficient normalization method employs

$$\overline{\mathbf{w}}(k+1) = \mathbf{w}(k) + \mathbf{g}(k) \tag{12}$$

$$\mathbf{w}(k+1) = \frac{\overline{\mathbf{w}}(k+1)}{\|\overline{\mathbf{w}}(k+1)\|} \tag{13}$$

which is a two-step update, to maintain $\|\mathbf{w}(k)\| = 1$ at each iteration. Equation (13) normalizes the length of $\mathbf{w}(k+1)$ without changing the direction of $\mathbf{w}(k+1)$ in n-dimensional space. To reduce computational complexity, we can often employ (12) with $\overline{\mathbf{w}}(k) = \mathbf{w}(k)$ for several iterations and invoke (13) infrequently. In addition, this method can be employed for any $\mu(k)$ and choice of norm, although a small value of $\mu(k)$ is usually required for stochastic gradient implementations.

If $\|\mathbf{w}\| = \|\mathbf{w}\|_2$ is chosen, (12) and (13) can be written compactly as

$$\mathbf{w}(k+1) = \frac{\mathbf{w}(k) + \mathbf{g}(k)}{\sqrt{1 + 2\mathbf{w}^T(k)\mathbf{g}(k) + \|\mathbf{g}(k)\|_2^2}}. \tag{14}$$

This update is also in the form of a rotation of $\mathbf{w}(k)$ in the direction of $\mathbf{h}_{\mathbf{g}}(k)$, where the angle of rotation is listed in the second entry of Table I.

Tangent Gradient Method [2]. The constraint $\|\mathbf{w}\| = 1$ restricts the space of parameter vectors to those that lie on the surface of an n-dimensional geometric object (e.g., a hypersphere, hyperpolyhedron, hypercube, etc.). Can the direction of $\mathbf{g}(k)$ be modified so that its integrated value lies on the constraint surface? The following theorem yields one possible solution to this problem, the proof of which appears in [18].

Theorem 1: Let $\|\mathbf{w}\|$ denote any differentiable vector norm, and let $\mathbf{g}(t)$ be any vector function with finite L_2-norm. Then

$$\frac{d\mathbf{w}(t)}{dt} = \mathbf{h}_{\mathbf{g}}(t) = \left(\mathbf{I} - \frac{\mathbf{v}(t)\mathbf{v}(t)^T}{\|\mathbf{v}(t)\|_2^2}\right)\mathbf{g}(t) \tag{15}$$

with $\|\mathbf{w}(0)\| = 1$ defines a vector function $\mathbf{w}(t)$ that satisfies $\|\mathbf{w}(t)\| = 1$ for all $t \geq 0$.

To obtain a useful algorithm from (15), substitute time differences for time differentials to obtain

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \mathbf{h}_{\mathbf{g}}(k) \\ &= \mathbf{w}(k) + \mathbf{g}(k) - \mathbf{v}(k)\frac{\mathbf{v}^T(k)\mathbf{g}(k)}{\|\mathbf{v}(k)\|^2}. \end{aligned} \tag{16}$$

This update does not guarantee that $\|\mathbf{w}(k)\| = 1$ for all k, however, and in fact

$$\|\mathbf{w}(k+1)\| \geq \|\mathbf{w}(k)\| \tag{17}$$

at each iteration if (16) is used for any valid vector norm [18]. Even so, updating $\mathbf{w}(k)$ using $\mathbf{h}_{\mathbf{g}}(k)$ instead of $\mathbf{g}(k)$ as in (11) largely decreases the rate at which $\|\mathbf{w}(k)\|$ deviates from $\|\mathbf{w}(k)\| = 1$. To stabilize this

TABLE II
SUMMARY OF ALGORITHMS FOR L_2-NORM CONSTRAINED ADAPTATION

Algorithm $\mathbf{w}_{new}$	*Complexity* $\times$	$+$	$\div$	$\sqrt{\cdot}$	cos()	*Stability Behavior*	*Stabilization Method*
$\mathbf{w}_{new} = \alpha\mathbf{w} + \mathbf{g}$, $\alpha = \sqrt{1 - \|\mathbf{g}\|_2^2 + [\mathbf{w}^T\mathbf{g}]^2} - \mathbf{w}^T\mathbf{g}$	$3n+1$	$3n+1$	0	1	0	Stable; $\|\mathbf{w}\|_2 = 1$	—
$\mathbf{w}_{new} = \dfrac{\mathbf{w} + \mathbf{g}}{\sqrt{1 + 2\mathbf{w}^T\mathbf{g} + \|\mathbf{g}\|_2^2}}$	$2n$	$2n-1$	1	1	0	Stable; $\|\mathbf{w}\|_2 = 1$	—
$\mathbf{w}_{new} = \left(1 - \dfrac{\mathbf{w}^T\mathbf{g}}{\|\mathbf{w}\|_2^2}\right)\mathbf{w} + \mathbf{g}$	$3n$	$3n-1$	1	0	0	Slow growth in $\|\mathbf{w}\|_2$	Set $\|\mathbf{w}\|_2 = 1$ infrequently
$\mathbf{w}_{new} = \cos\theta\mathbf{w} + \dfrac{\sin\theta}{\theta}\mathbf{h}_\mathbf{g}$, $\mathbf{h}_\mathbf{g} = \mathbf{g} - \mathbf{w}\mathbf{w}^T\mathbf{g}, \theta = \|\mathbf{h}_\mathbf{g}\|_2$	$5n$	$4n-2$	1	1	2	Stable; $\|\mathbf{w}\|_2 = 1$	—
$\mathbf{w}_{new} = \left(1 - \mathbf{w}^T\mathbf{g}\right)\mathbf{w} + \|\mathbf{w}\|_2^2\mathbf{g}$	$4n$	$3n-1$	0	0	0	Accelerated growth in $\|\mathbf{w}\|_2$	Set $\|\mathbf{w}\|_2 = 1$ periodically
$\mathbf{w}_{new} = \left(1 - \mathbf{w}^T\mathbf{g}\right)\mathbf{w} + \mathbf{g}$	$2n$	$2n$	0	0	0	$\|\mathbf{w}\|_2 \approx 1$ if $\mathbf{w}^T\mathbf{g} > 0$	If $\mathbf{w}^T\mathbf{g} < 0$, set $\|\mathbf{w}\|_2 = 1$ when $\|\mathbf{w}\|_2^2 > C$, $1.1 \le C \le 1.5$
$\mathbf{w}_{new} = \left(1 - \mathbf{w}^T\mathbf{g}\right)\mathbf{w} + \|\mathbf{w}\|_2^4\mathbf{g}$	$4n+1$	$3n-1$	0	0	0	$\|\mathbf{w}\|_2 \approx 1$ if $\mathbf{w}^T\mathbf{g} < 0$	—

update, we still need to infrequently normalize the length of $\mathbf{w}(k)$ via (13).

In the case of the L_2 norm, this algorithm is

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{g}(k) - \mathbf{w}(k)\frac{\mathbf{w}^T(k)\mathbf{g}(k)}{\|\mathbf{w}(k)\|_2^2} \tag{18}$$

which is also the natural gradient algorithm on the unit hypersphere [17]. A calculation shows that

$$\|\mathbf{w}(k+1)\|_2^2 = \|\mathbf{w}(k)\|_2^2 + \|\mathbf{h}_\mathbf{g}(k)\|_2^2 \tag{19}$$

and since $\|\mathbf{h}_\mathbf{g}(k)\|_2$ is of $O(\mu(k))$, the growth of $\|\mathbf{w}(k)\|_2^2$ is linear in $\mu^2(k)$. If the length of $\mathbf{w}(k)$ is normalized at each iteration, this update is also in the form of a rotation in the direction of $\mathbf{h}_\mathbf{g}(k)$, and the angle of rotation is listed in the third entry of Table I.

True Gradient Method [7], [16]. Each of the previous algorithms approximates the following true gradient adaptation procedure for (1) and (2).

i) Calculate the tangent gradient $\mathbf{h}_\mathbf{g}(k)$ in (5).
ii) Move a distance $\|\mathbf{h}_\mathbf{g}(k)\|_2$ along a geodesic of the constraint surface in the direction of $\mathbf{h}_\mathbf{g}(k)$.

A geodesic is a curve on the constraint surface that connects two arbitrary points by an arc of shortest length. Implementing this procedure for a given norm constraint requires knowledge of the equations of motion on the constraint surface.

When $\|\mathbf{w}(k)\|_2 = 1$ is imposed, updating $\mathbf{w}(k)$ amounts to rotating $\mathbf{w}(k)$ by an angle $\theta(k)$. For any unit vector $\mathbf{u}(k)$ that is perpendicular to $\mathbf{w}(k)$, the update

$$\mathbf{w}(k+1) = \cos(\theta(k))\mathbf{w}(k) + \sin(\theta(k))\mathbf{u}(k) \tag{20}$$

rotates $\mathbf{w}(k+1)$ by an angle $\theta(k)$ in the direction of $\mathbf{u}(k)$. For gradient adaptation, we choose

$$\mathbf{u}(k) = \frac{\mathbf{h}_\mathbf{g}(k)}{\theta(k)} = \frac{1}{\theta(k)}\left(\mathbf{g}(k) - \mathbf{w}(k)\mathbf{w}^T(k)\mathbf{g}(k)\right) \tag{21}$$

where the form of $\theta(k) = \|\mathbf{h}_\mathbf{g}(k)\|_2$ is given in the last entry of Table I. Note that when $\mu(k)$ is small, $\tan(\theta(k)) \approx \sin(\theta(k)) \approx \theta(k)$, yielding similar angles of rotation for all four methods.

III. IMPLEMENTATION ISSUES FOR L_2-NORM CONSTRAINED METHODS

We consider the computational complexities and numerical stabilities of L_2-norm constrained gradient approaches in this section, as the L_2-norm constraint is the most popular for practical applications. To illustrate the salient issues involved in algorithm design, we will focus on

$$\mathcal{J}(\mathbf{w}) = \pm\frac{1}{p}E\{|y(k)|^p\} \tag{22}$$

as an instantaneous cost function, where $y(k) = \mathbf{w}^T(k)\mathbf{x}(k)$, $\mathbf{x}(k)$ is a discrete-time vector random process, and p is a positive integer not equal to 2. This cost function arises in certain formulations of independent component analysis (ICA), blind source separation, and blind deconvolution [13]–[15]. In this case

$$\mathbf{g}(k) = \pm\mu(k)|y(k)|^{p-2}y(k)\mathbf{x}(k) \quad \text{and}$$
$$\mathbf{w}^T(k)\mathbf{g}(k) = \pm\mu(k)|y(k)|^p \tag{23}$$

such that the sign of $\mathbf{w}^T(k)\mathbf{g}(k)$ does not change for all k.

The first four rows of Table II list the complexities of each of the algorithms in (9)–(10), (14), (18), (20), and (21) according to the number and type of operations required, neglecting those operations needed for calculating $\mathbf{g}(k)$. The Lagrange, coefficient normalization, and tangent gradient methods are the simplest, whereas the true gradient method is significantly more complicated. All of these methods require operations other than multiply/adds to implement, making them more difficult to implement on real-time signal processing devices that are optimized for multiply/add calculations. Note that the structure of $\mathbf{g}(k)$ can often be exploited to further reduce each algorithm's complexity, e.g., by using (23) to compute $\mathbf{w}^T(k)\mathbf{g}(k)$ in the case of (22).

We now consider simplifications that yield similar-behaving algorithms with reduced complexities. Since all four algorithms have equivalent behavior up to $O(\mu^2(k))$, we only consider modifications of one approach—the tangent gradient method in (18). The modified versions are as follows.

Modification #1 [11]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \|\mathbf{w}(k)\|_2^2\mathbf{g}(k) - \mathbf{w}(k)\mathbf{w}^T(k)\mathbf{g}(k). \tag{24}$$

Modification #2 [6]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{g}(k) - \mathbf{w}(k)\mathbf{w}^T(k)\mathbf{g}(k). \tag{25}$$

Modification #3 [12], [15]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \|\mathbf{w}(k)\|_2^4\mathbf{g}(k) - \mathbf{w}(k)\mathbf{w}^T(k)\mathbf{g}(k). \tag{26}$$

If $\|\mathbf{w}(k)\|_2^2 = 1$, all three modified methods have similar behaviors to that of (18); however, the numerical properties of the algorithms in the radial dimension associated with the length of $\mathbf{w}(k)$ are quite different, as we will show. For simplicity of discussion, we define

$$c(k) = \|\mathbf{w}(k)\|_2^2 - 1. \tag{27}$$

If $c(k)$ experiences unmitigated growth, the associated algorithm is numerically unstable.

Numerical Stability of (24): Premultiplying both sides of (24) by their transposes yields

$$\|\mathbf{w}(k+1)\|_2^2 = \left[1 + \|\mathbf{w}(k)\|_2^2\mathbf{g}^T(k)\left(\mathbf{I} - \frac{\mathbf{w}(k)\mathbf{w}^T(k)}{\|\mathbf{w}(k)\|_2^2}\right)\mathbf{g}(k)\right] \times \|\mathbf{w}(k)\|_2^2. \tag{28}$$

The matrix in large parentheses on the right-hand side of (28) is a projection matrix. Hence, as long as $\mathbf{g}(k)$ is not collinear with $\mathbf{w}(k)$ and $\|\mathbf{g}(k)\|_2 > 0$, then $\|\mathbf{w}(k+1)\|_2 > \|\mathbf{w}(k)\|_2$ if $\|\mathbf{w}(0)\|_2 = 1$, i.e., numerical instability. Furthermore, since $\|\mathbf{w}(k)\|_2^2$ appears in the factor in brackets on the right-hand side of (28), (24) causes accelerated growth in $\|\mathbf{w}(k)\|_2^2$ independently of the form of $\mathcal{J}(\mathbf{w})$.

Numerical Stability of (25): Premultiplying both sides of (25) by their transposes, subtracting one from both sides, and rearranging terms, we obtain

$$c(k+1) = \left[1 - 2\mathbf{w}^T(k)\mathbf{g}(k)\right]c(k) + \left\|\mathbf{g}(k) - \mathbf{w}(k)\mathbf{w}^T(k)\mathbf{g}(k)\right\|_2^2. \tag{29}$$

If $\|\mathbf{w}(0)\|_2 = 1$, then we have $c(k) > 0$ for all k if $\mathbf{g}(k) \neq \mathbf{w}(0)\mathbf{w}^T(0)\mathbf{g}(k)$. Since the second term on the right-hand side of (29) is non-negative, we require

$$\mathbf{w}^T(k)\mathbf{g}(k) > 0 \tag{30}$$

for the numerical stability of (25). Note that (30) can often be verified for a particular $\mathcal{J}(\mathbf{w})$. For example, if (22) is chosen with a positive sign, then (30) is always true, making this algorithm appropriate for constrained maximization of $p^{-1}E\{|y(k)|^p\}$. Conversely, the algorithm fails when $p^{-1}E\{|y(k)|^p\}$ is being minimized for all $\|\mathbf{w}\|_2 = 1$. This result is what justifies {(23), (25)} for use in maximum-kurtosis ICA and principal component analysis tasks, and it also explains why this algorithm fails for minimum-kurtosis ICA and minor component analysis tasks [12], [15].

Numerical Stability of (26): Performing a similar analysis of (26) as used previously, we obtain an update for $c(k)$ as

$$c(k+1) = \left[1 + 2\|\mathbf{w}(k)\|_2^2\mathbf{w}^T(k)\mathbf{g}(k)\right]c(k) + \left\|\|\mathbf{w}(k)\|_2^4\mathbf{g}(k) - \mathbf{w}(k)\mathbf{w}^T(k)\mathbf{g}(k)\right\|_2^2. \tag{31}$$

In this case, if

$$\mathbf{w}^T(k)\mathbf{g}(k) < 0 \tag{32}$$

then (26) is numerically stable. For minimum-kurtosis ICA ($p = 4$) tasks, choosing $\mathbf{g}(k) = -\mu(k)|y(k)|^{p-2}y(k)\mathbf{x}(k)$ causes this algorithm to perform in a stable fashion.

Other Approaches: Of the three methods in (24)–(26), (25) is the simplest, requiring $2n$ multiply/adds at each iteration. When $\mathbf{w}^T(k)\mathbf{g}(k) < 0$, however, this method is numerically unstable. As an alternative to (26), we can monitor the stability behavior of (25) and rescue the system from instability just prior to its occurence. It can be shown from (29) that $\|\mathbf{w}(k)\|_2^2 > 2$ is an indicator of the onset of sudden divergence of (25) [18]. Moreover, simulations of (25) in several situations indicate that $\|\mathbf{w}(k)\|_2^2 > 2$ provides a reliable divergence indicator when $\mathbf{w}^T(k)\mathbf{g}(k) < 0$. We can therefore monitor the value of $\|\mathbf{w}(k)\|_2^2$ and renormalize $\mathbf{w}(k)$ to unit length when $\|\mathbf{w}(k)\|_2^2 > C$, where $1 < C < 2$. Simulations indicate that values of C in the range $1.1 \leq C \leq 1.5$ often yield good performance for typical problems and choices of $\mu(k)$. In addition, since $\|\mathbf{w}(k)\|_2^2$ tends to grow slowly for small values of $\mu(k)$, the test $\|\mathbf{w}(k)\|_2^2 > C$ need only be performed at every Lth iteration, where $L \gg 1$, such that the complexity associated with this rescue method can be minimized. Further details regarding this approach, along with alternative reduced parameterization gradient methods for (1) and (2), can be found in [18].

IV. Simulations

We now explore the behaviors of the algorithms in (18), (25), and (26) in a single-component ICA task via MATLAB simulations. Let

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) \tag{33}$$

where $\mathbf{A}$ is an $(n \times n)$ constant mixing matrix, and $\mathbf{s}(k) = [s_1(k) \ldots s_n(k)]^T$ contains unobservable independent components. If $\mathbf{A}\mathbf{A}^T = \mathbf{I}$, then maximizing $\mathcal{J}(\mathbf{w}) = -0.25E\{|y(k)|^4\}$ for $y(k) = \mathbf{w}^T(k)\mathbf{x}(k)$ and $\|\mathbf{w}(k)\|_2 = 1$ results in a negative-kurtosis signal in $y(k)$ [14], [15]. In practice, we can guarantee $\mathbf{A}\mathbf{A}^T = \mathbf{I}$ by prewhitening the signal measurements using simple adaptive procedures [9], [19]. We can then choose $\mathbf{g}(k) = -\mu(k)|y(k)|^2y(k)\mathbf{x}(k)$ for any of the previously discussed algorithms to obtain a candidate algorithm for this task.

For our simulations, we generate $\mathbf{s}(k) = [s_1(k)\ s_2(k)\ s_3(k)]^T$, where $s_1(k)$ is an i.i.d. binary-$\{\pm 1\}$-distributed signal, and $s_2(k)$ and $s_3(k)$ are i.i.d. Laplacian-distributed signals with p.d.f. $p_s(s) = e^{-\sqrt{2}|s|}/\sqrt{2}$. The mixing matrix $\mathbf{A}$ is chosen to be the eigenvector matrix of

$$\mathbf{R}_{\mathbf{xx}} = \begin{bmatrix} 0.9 & 0.4 & 0.7 \\ 0.4 & 0.3 & 0.5 \\ 0.7 & 0.5 & 1.0 \end{bmatrix} \tag{34}$$

such that measurement prewhitening is not needed. For each algorithm, 100 simulations have been run, and the average values of the performance factors

$$\rho(k) = \|\mathbf{c}_1(k)\|_2^2/\|\mathbf{c}_2(k)\|_2^2 \quad \text{and} \quad \eta(k) = \left[\|\mathbf{w}(k)\|_2^2 - 1\right]^2 \tag{35}$$

have been computed, where $\mathbf{c}_i(k) = \mathbf{E}_i^T\mathbf{w}(k)$ and $\mathbf{E}_1$ and $\mathbf{E}_2$ contain the 2-D and 1-D subspaces corresponding to the signal directions of the Laplacian and binary sources.

Fig. 1(a) shows the evolution of $\rho(k)$ for the tangent gradient method in (18), the simplified method in (25) with stabilization, and the self-normalized method in (26) for this task, where $\mu(k) = 0.001$, $L = 20$, and $C = 1.1$. All three algorithms are successful at extracting the binary source from the linear mixture. Fig. 1(b) shows the average evo-

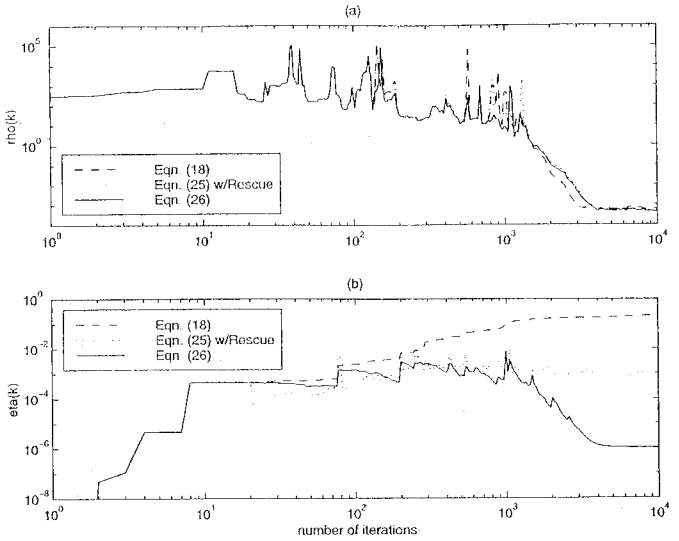

Fig. 1. Average performances of the various algorithms in the minimum-kurtosis ICA task. (a) Evolutions of $\rho(k)$. (b) Evolutions of $\eta(k)$.

lution of $\eta(k)$ for the three methods, where the unmitigated growth in $\|\mathbf{w}(k)\|_2$ for the tangent gradient method is clearly evident. In contrast, the stabilization procedure used for (25) maintains this algorithm's stable behavior, and (26) performs in a stable, self-normalizing manner without such intervention.

V. Conclusion

In this correspondence, we have presented an overview of algorithms that adjust a parameter vector to minimize or maximize a chosen cost function under a unit-norm parameter vector constraint. Particular attention has been paid both to methods that guarantee the unit-norm constraint at each iteration and to methods that maintain $\|\mathbf{w}(k)\|_2 \approx 1$ over time. Simulations verify the useful behavior of the schemes for independent component analysis. Some extensions of these results to multiple dimensions can be found in [12].

References

[1] R. Courant and D. Hilbert, *Methods of Mathematical Physics*. New York: Interscience, 1953, vol. I.

[2] T. P. Krasulina, "Method of stochastic approximation in the determination of the largest eigenvalue of the mathematical expectation of random matrices," *Automat. Remote Contr.*, vol. 2, pp. 215–221, 1970.

[3] N. L. Owsley, "Adaptive data orthonormalization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Tulsa, OK, Apr. 1978, pp. 109–112.

[4] P. A. Thompson, "An adaptive spectral analysis technique for unbiased frequency estimation in the presence of white noise," in *Proc. 13th Asilomar Conf. Circ., Syst., Comput.*, Pacific Grove, CA, Nov. 1979, pp. 529–533.

[5] V. U. Reddy, B. Egardt, and T. Kailath, "Least squares type algorithm for adaptive implementation of Pisarenko's harmonic retrieval method," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 399–405, June 1982.

[6] E. Oja, "A simplified neural model as a principal component analyzer," *J. Math. Biol.*, vol. 15, pp. 267–273, 1982.

[7] D. R. Fuhrmann and B. Liu, "An iterative algorithm for locating the minimal eigenvector of a symmetric matrix," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, 1984, pp. 45.8.1–45.8.4.

[8] L. Wang and J. Karhunen, "A unified neural bigradient algorithm for robust PCA and MCA," *Int. J. Neural Syst.*, vol. 7, no. 1, pp. 53–67, Mar. 1996.

[9] K. I. Diamantaras and S.-Y. Kung, *Principal Component Neural Networks: Theory and Applications*. New York: Wiley, 1996.

[10] V. Solo and X. Kong, "Performance analysis of adaptive eigenanalysis algorithms," *IEEE Trans. Signal Processing*, vol. 46, pp. 636–646, Mar. 1998.

[11] T.-P. Chen, S. Amari, and Q. Lin, "A unified algorithm for principal and minor components extraction," *Neural Networks*, vol. 11, pp. 385–390, Apr. 1998.

[12] S. C. Douglas, S.-Y. Kung, and S. Amari, "A self-stabilized minor subspace rule," *IEEE Signal Processing Lett.*, vol. 5, pp. 328–330, Dec. 1998.

[13] P. Comon, "Independent component analysis: A new concept?," *Signal Process.*, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[14] A. Hyvarinen and E. Oja, "Independent component analysis by general nonlinear Hebbian-like learning rules," *Signal Process.*, vol. 64, no. 3, pp. 301–313, Feb. 1998.

[15] S. C. Douglas and S.-Y. Kung, "KuicNet algorithms for blind deconvolution," in *Proc. IEEE Workshop Neural Networks Signal Process.*, Cambridge, U.K., Aug. 1998, pp. 3–12.

[16] S. T. Smith, "Geometric optimization methods for adaptive filtering," Ph.D. dissertation, Harvard Univ., Cambridge, MA, 1993.

[17] S. C. Douglas and S. Amari, "Natural gradient adaptation," in *Unsupervised Adaptive Filtering, Vol I: Blind Source Separation*, S. Haykin, Ed. New York: Wiley, 1999, ch. 2.

[18] S. C. Douglas, S. Amari, and S.-Y. Kung, "Gradient adaptation with unit-norm constraints," Dept. Elect. Eng., Southern Methodist Univ., Dallas, TX, Tech. Rep. EE-99-003, Feb. 1999.

[19] S. C. Douglas and A. Cichocki, "Neural networks for blind decorrelation of signals," *IEEE Trans. Signal Processing*, vol. 45, pp. 2829–2842, Nov. 1997.