COMPLETE TRAINING ANALYSIS OF FEEDBACK ARCHITECTURE NETWORKS THAT PERFORM BLIND SOURCE SEPARATION AND DECONVOLUTION

Nikos A. Kanlis, Jonathan Z. Simon, and Shihab A. Shamma

University of Maryland
Institute for Systems Research &
Electrical and Computer Engineering Department
Neural Systems Lab, A.V.Williams Bldg, Room 2202
College Park, MD 20742
nkanlis,jzsimon,sas@eng.umd.edu

ABSTRACT

In this paper we address the difficult problem of separating multiple speakers in a real-world situation, were the recordings are not just instantaneous mixtures, but rather mixtures of filtered versions of the sources. The enhancement over the approaches already presented by other researchers is that our model allows direct-path, zero-delayed versions of all the sources to be present in each one of the mixtures (a difficult approach because it introduces recursiveness in the model, but a closer to the reality one). The update rules are all derived in matrix form (suitable for computing environments, e.g Matlab), with special attention to the diagonals of those matrices, in order to avoid "temporal whitening" at the output. Extending those update rules to ones based on "natural" gradient is also addressed.

1. INTRODUCTION

Blind separation of independent sources (BSS) is a growing area with potential in many applications (enhancement of biomedical images and signals, efficient bandwidth usage in wireless communications, noise suppression in sonar and radar signals, etc). Recent research (equivariant adaptive algorithm by Cardoso and Laheld, 1996 [8], entropy maximization by Bell and Sejnowski, 1995 [6], natural gradient approach by Amari *et al.*, 1996 [2],1998 [1]), has produced robust and fast solutions to the problem of blind separation of instantaneous mixtures, namely the case:

$$\mathbf{x}(k) = A \cdot \mathbf{s}(k) \tag{1}$$

where $\mathbf{x}(k) = [x_1(k) \dots x_N(k)]^T$ is the N-dimensional discrete time vector of the recordings at time k, $\mathbf{s}(k) = [s_1(k) \dots s_M(k)]^T$ is the M-dimensional vector of the source signals also at time k, and A is the $(N \times M)$-dimensional matrix of mixing coefficients. The prevailing technique employed to blindly achieve the desired signal separation is the one that adjusts the coefficients in a single-layer neural network so that the entropy at the output is maximized (equivalently, mutual information between outputs is being minimized, Zhang *et al.* [28], or mutual information between the inputs and the outputs is being maximized, Bell and Sejnowski [7], [5]). Since any mixture of statistically independent signals has lower entropy that the non-mixed versions, maximization of entropy leads to source separation. Using such information theoretic principles to solve the inverse problem is extremely versatile because it does not rely on modeling the underlying physical phenomena, Cardoso and Laheld [8]. For real world situations though, the model of instantaneous mixing is not a good one. Not only in speech, but also in other applications (sonar, EEGs, etc.), delayed and filtered versions of the sources (*e.g.* due to echoes) get recorded along with the direct-path waves, thus suggesting a mixing architecture more of the form:

$$\mathbf{x}(k) = \sum_{\mathbf{l=0}}^{L} A_l \cdot \mathbf{s}(k-l) \tag{2}$$

where the various A_l's are the $(N \times M)$-dimensional matrices of mixing coefficients at different time lags extending up to lag L. This scheme is equivalent to having each one of the sources $s_i(k)$ being passed through an FIR filter $\mathbf{A}_{ji}(z)$ of $L+1$ taps ($\mathbf{A}_{ji}(z) = \sum_{\mathbf{l=0}}^{L} [A_l]_{ji} \cdot z^{-l}$ in the frequency domain), before being picked up by the microphone j. Thus eq.(2) written in the frequency domain for the 2-sources 2 microphones case (Figure 1), looks as:

$$\begin{bmatrix} \mathbf{x}_1(z) \\ \mathbf{x}_2(z) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}(z) & \mathbf{A}_{12}(z) \\ \mathbf{A}_{21}(z) & \mathbf{A}_{22}(z) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{s}_1(z) \\ \mathbf{s}_2(z) \end{bmatrix} \tag{3}$$

Significant work has been done in this case, either in the time-domain pursuing the feedforward architecture case by Amari, Cichocki and colleagues [9, 4, 3, 11, 10], where FIR filters acting directly on the recordings are being learned, or using a more intuitive feedback architecture as in the works of Torkkola, [24, 25], and Lee *et al.* later, [19], or in the frequency-domain where time-convolutions are just multiplications and the whole problem collapses to the instantaneous mixing/unmixing one, Smaragdis [23].

This work was supported by the Office of Naval Research, MURI# N00014-97-1-0501

2. ARCHITECTURE

The principle behind the time-domain approaches is the same as for the instantaneous mixing/unmixing case: Maximization of the entropy at the output of the network leads to separation AND deconvolution, because as with the mixing of different statistically independent sources, redundant delayed versions of the same signal result in less entropy overall. Due to the nature of this principle, one major drawback that the feedforward architecture suffers is that it introduces "temporal whitening" on the recovered sources. Since every signal has inherent short-term dependencies among its samples [26](up to some 5-6 msecs for speech signals, translating to some 40-50 samples for a 8KHz-sampled signal), maximization of entropy at the output removes those dependencies also, in addition to all the rest. The result is "whitened" signals, signals that have flat spectrum, although the phase information is being preserved. The blind separation network architecture that we propose is an extension of the one introduced by Torkkola [24]. In an effort to avoid the temporal whitening, he modeled the solution by using a *feedback* architecture (Figure 1 for the two sources - two microphones case) where estimates of the sources are fed back in the network, on all branches, except the ones that each particular source is being estimated out of. With this architecture though, one should be content with obtaining what each sensor would observe in the absence of the interfering sources without any other distorting effects [24], namely $\mathbf{u}_i(z) = \mathbf{A}_{ii}(z)\mathbf{s}_i(z)$.

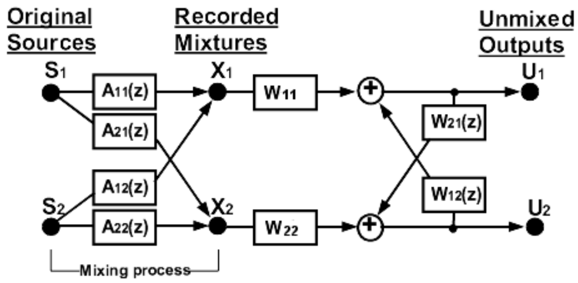

Figure 1: Feedback architecture for separation and deconvolution for the 2 sources - 2 microphones case.

In principle, IIR filters can solve the problem (as long as the direct paths are "good", [27], meaning minimum-phase, although non-causal filters can always be used to overcome this problem [14]), but the particular architecture proposed by Torkkola only models the case where each one of the microphones is directly associated with, and is closer to, one of the sources (e.g. in a conference room with a microphone allocated to each one of the speakers). His model fails in the general case through the lack of feedback cross-weights for delay $l = 0$, thus not being able to provide solution when two sources arrive on two microphones without delay relative to each other. Incorporating zero delay coefficients into those FIR filters though, is not an easy task because of the recursiveness it introduces: Estimates of the signals for the same time instant appear on both sides of the model equations. Denoting by A_l^1 the matrix having diagonal elements equal to those of A_l, and all the rest equal to 0 ($A_l^1 = diag(diag(A_l))$ in Matlab notation), and by A_l^0 the matrix equal to A_l but with its diagonal 0 ($A_l^0 = A_l - A_l^1$), we can rewrite eq.(2) as:

$$\mathbf{x}(k) = \sum_{l=0}^{L} A_l^1 \cdot \mathbf{s}(k-l) + \sum_{l=0}^{L} A_l^0 \cdot \mathbf{s}(k-l)$$

$$\Longrightarrow \sum_{l=0}^{L} A_l^1 \cdot \mathbf{s}(k-l) = \mathbf{x}(k) - \sum_{l=0}^{L} A_l^0 \cdot \mathbf{s}(k-l) \quad (4)$$

or written in the z-domain for the 2x2 case:

$$\begin{bmatrix} \mathbf{A}_{11}(z)\mathbf{s}_1(z) \\ \mathbf{A}_{22}(z)\mathbf{s}_2(z) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1(z) \\ \mathbf{x}_2(z) \end{bmatrix} - \begin{bmatrix} 0 & \mathbf{A}_{12}(z) \\ \mathbf{A}_{21}(z) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1(z) \\ \mathbf{s}_2(z) \end{bmatrix} \quad (5)$$

The terms on the left hand sides of eq.(4) and eq.(5) are what the microphones would have picked up for each source (without the artifact of whitening), if all the other sources and interferences were absent. Since we are after this solution, the best model for the estimates $\mathbf{u}$(k) of the independent sources $\mathbf{s}$(k) is:

$$\mathbf{u}(k) = W \cdot \mathbf{x}(k) + \sum_{l=0}^{L'} W_l^0 \cdot \mathbf{u}(k-l) \quad (6)$$

or in the frequency domain for the 2x2 case:

$$\begin{bmatrix} \mathbf{u}_1(z) \\ \mathbf{u}_2(z) \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{x}_1(z) \\ \mathbf{x}_2(z) \end{bmatrix} + \begin{bmatrix} 0 & \mathbf{W}_{12}(z) \\ \mathbf{W}_{21}(z) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1(z) \\ \mathbf{u}_2(z) \end{bmatrix} \quad (7)$$

where $\mathbf{W}$ is a diagonal matrix that has to be learned. It is not obvious from eq.(5) why such a matrix is necessary, since it seems without it the algorithm would be just fine. It is there because the estimated components need to match the dynamic range of the non-linearity at the output of the network, and it is required to be diagonal because each recording has to be scaled individually, with no "untreated" portions of other channels leaking in, which would lead to temporal whitening. The superscript on the matrices W_l^0 means that the diagonal of those matrices is and must remain zero to avoid temporal whitening too. As long as the direct-path filters are "good", the solution for eq.(5) is:

$$\mathbf{u}_1(z) = [\mathbf{W}]_{11} \cdot \mathbf{A}_{11}(z) \cdot \mathbf{s}_1(z)$$
$$\mathbf{u}_2(z) = [\mathbf{W}]_{22} \cdot \mathbf{A}_{22}(z) \cdot \mathbf{s}_2(z)$$

and

$$\mathbf{W}_{12}(z) = -[\mathbf{W}]_{11} \cdot \mathbf{A}_{12}(z) \cdot \left([\mathbf{W}]_{22} \cdot \mathbf{A}_{22}(z)\right)^{-1}$$
$$\mathbf{W}_{21}(z) = -[\mathbf{W}]_{22} \cdot \mathbf{A}_{21}(z) \cdot \left([\mathbf{W}]_{11} \cdot \mathbf{A}_{11}(z)\right)^{-1}$$

3. UPDATE RULES

The challenge at this point on the update of the weights of our neural network is twofold: How do we maximize the entropy at the output of our network since the output vector

$\mathbf{u}(k)$ appears in both sides of eq.(6)? How do we write the updates of the weights in matrix form, keeping the diagonal of the matrices zero after the update? In order to deal with the first challenge, we solve eq.(6) for the $\mathbf{u}$(k) vector:

$$\mathbf{u}(k) = (I - W_0^0)^{-1} \left[W \cdot \mathbf{x}(k) + \sum_{\mathbf{l=1}}^{L'} W_l^0 \cdot \mathbf{u}(k-l) \right]$$

or, by substituting $\hat{W} = (I - W_0^0)^{-1}$:

$$\mathbf{u}(k) = \hat{W} \cdot W \cdot \mathbf{x}(k) + \sum_{\mathbf{l=1}}^{L'} \hat{W} \cdot W_l^0 \cdot \mathbf{u}(k-l) \qquad (8)$$

So, the matrices that have to be learned are $\hat{W}$ and $W_l^0, L' \geq l \geq 1$. The most popular learning method in the general nonlinear optimization framework is the stochastic gradient updating the weights proportionally to the derivative of the entropy with respect to those weights. The entropy of the output can be written as the expectation of the log probability density function of the output. Since $f_{\mathbf{u}}(\mathbf{u}) = f_{\mathbf{x}}(\mathbf{x})/|J|$ [21, eq.5-6], where J is the Jacobian of the whole system, we get:

$$\begin{aligned} H(\mathbf{u}) &= -E[\ln(f_{\mathbf{u}}(\mathbf{u}))] \\ &= -E[\ln(f_{\mathbf{x}}(\mathbf{x}))/|J|)] \\ &= -E[\ln(f_{\mathbf{x}}(\mathbf{x}))] + E[\ln|J|] \end{aligned} \qquad (9)$$

So, maximization of $H(\mathbf{u})$ is equivalent to maximization of $E[\ln|J|]$, since $f_{\mathbf{x}}(\mathbf{x})$ is set and cannot change with any choice of the network parameters. Using this and the detailed stochastic gradient analysis of our network in Appendix A, the update rules are:

$$\Delta W \propto W^{-T} + \text{diag}^1[\hat{W}^T \cdot \hat{\mathbf{y}}(k) \cdot \mathbf{x}^T(k)] \qquad (10)$$

$$\Delta \hat{W} \propto \left[I + \hat{\mathbf{y}}(k) \cdot \mathbf{u}^T(k)\right] \cdot [\hat{W}]^{-T} \qquad (11)$$

$$\Delta W_l \propto \text{diag}^0[\hat{W}^T \cdot \hat{\mathbf{y}}(k) \cdot \mathbf{u}^T(k-l)] \qquad (12)$$

where $k \geq 0$, $L' \geq l \geq 1$, $diag^0[\ldots]$ means that the diagonal of the argument matrix is set to zero, $diag^1[\ldots]$ means that the non-diagonal elements are set to zero, and $\hat{\mathbf{y}}(k)$ is a vector depending on the nonlinearity $y = y(u)$ operating at the output of the network. The optimal choice for the nonlinearity that assures local convergence with the fastest rate, would be the proportional to the *cdf* of the sources [6, 5], [8, 2, 22]. In the optimum of the cases, a different non-linearity should be used on each output branch, matching the *pdf* of the source separated in that output. In mathematical detail, the elements of the $\hat{\mathbf{y}}(k)$ are

$$\hat{y}_i(k) = \frac{\partial y_i'(k)}{\partial y_i(k)} = \frac{\partial}{\partial y_i(k)} \frac{\partial y_i(k)}{\partial u_i(k)} \approx \frac{\partial f_{s_i}(u_i)}{\partial F_{s_i}(u_i)} = \frac{f_{s_i}'(u_i)}{f_{s_i}(u_i)}$$

where f_{s_i} is the *pdf* and F_{s_i} the *cdf* of the source s_i. Many selections of those functions emerged in the literature, all of them initially concentrated on how to separate sources with super- or sub-Gaussian characteristics. For example Bell, Sejnowski and T-W Lee [6, 17], used the logistic function:

$$y_i(u_i) = (1 + e^{-u_i})^{-1} \Rightarrow \hat{y}_i(u_i) = 1 - 2 \cdot u_i \qquad (13)$$

as an approximation to the *cdf* of the audio sources, because this non-linear function is suitable for speech-like super-Gaussian sources. Amari *et al.* [3, 11] used $\hat{y}_i(u_i) = |u_i|^2 \cdot u_i$ for sub-Gaussian, negative kurtosis signals (like typical complex-valued baseband digital communication signals), and $\hat{y}_i(u_i) = \tanh(\gamma u_i), \gamma > 2$ for super-Gaussian, positive kurtosis signals. Although, suboptimal choices for those non-linearities still allow the algorithm to perform separation and deconvolution for the case of speech sources, the selection is critical other cases like analysis of EEG and MEG recordings. The signals generated in these cases by the neurons in the brain have distributions that are multimodal [16], and therefore more flexible non-linearities should be employed. An elegant way of parameterizing the non-linearity depending whether a sub- or super-Gaussian source is being separated in a given branch of the network, has been proposed by Girolami *et al.* [12, 13] by choosing negentropy as a projection pursuit index. This parameterization led to the formulation of the so-called Extended ICA Algorithm [18], that switches the selection of the non-linearity depending on the sign of a stability criterion, namely:

$$\begin{aligned} \hat{y}_i(k) = \; & -z_i \cdot \tanh(u_i) - u_i \\ \text{where} \; & \begin{cases} z_i = 1 & \text{when} \quad u_i: \quad \text{super-Gaussian} \\ z_i = -1 & \text{when} \quad u_i: \quad \text{sub-Gaussian} \end{cases} \end{aligned} \qquad (14)$$

and the decision on the super/sub-Gaussianity is made according to the following criterion:

$$z_i = \text{sgn}[E\{sech^2(u_i)\} \cdot E\{u_i^2\} - E\{tanh(u_i) \cdot u_i\}]$$

That parameterized version of the non-linearity, allowing separation with the use of only one neural network when both super and sub-Gaussian sources are present, is suitable for analysis of signals originating from the brain (e.g. has been used successfully on EEG recordings by Makeig et. al. [15]). Although we intend to apply the algorithm to process EEGs recorded from the surface of auditory cortex of ferrets [20] and that selection of the non-linearity would be the most appropriate, for the scope of this publication that speech signals are only concerned, we will limit ourselves to the logistic function of eq.(13).

Looking back to the update rule of $\hat{W}$ (eq.(11)), some more manipulation is necessary in order to derive an update for the more important matrix ${W_0}^0$, because during the training of our network we should use eq.(6) and not eq.(8) for the sources' estimation. The multiplication of $\hat{W}$ with matrices W_l on eq.(8) leads to temporal whitening, because their product does not preserve the zero diagonal requirement (setting the diagonal of the product equal to zero is not an option here!). It is tempting on the other hand at this point, to utilize the relationship between ${W_0}^0$ and $\hat{W}$ leading to $\Delta {W_0}^0 = -(\hat{W})^{-1} \Delta \hat{W} (\hat{W})^{-1}$ (as in [17]), but that model is numerically unstable. The indeterminacy of the algorithm up to scaling mentioned above has as a result, a scaling factor to be shared for each extracted source, among the respective row of W and column of $\hat{W}$, which breaks the requirement that the inverse of $\hat{W}$ have unit diagonal elements. The situation can be circumvented by the following manipulation:

$$\begin{aligned}\hat{W} &= [[\hat{W}]^{-1}]^{-1} =: [K]^{-1} = [K^1 \cdot ([K^1]^{-1} \cdot K)]^{-1} \\ &= [K^1 \cdot (I + [K^1]^{-1} \cdot K^0)]^{-1} \\ &= (I + \underbrace{[K^1]^{-1} \cdot K^0}_{-W_0{}^0})^{-1} \cdot \underbrace{[K^1]^{-1}}_{D} \\ &= (I - W_0{}^0)^{-1} \cdot D \end{aligned} \tag{15}$$

With this decomposition of $\hat{W}$, a diagonal matrix, namely

$$D = \mathrm{inv}(\mathrm{diag}^1(\mathrm{inv}(\hat{W}))) \tag{16}$$

multiplies W and W_l on eq.(8) on the left, thus preserving the requirements posed to the elements of those matrices, and

$$W_0{}^0 = -D \cdot \mathrm{diag}^0(\mathrm{inv}(\hat{W})) \tag{17}$$

is used on the estimation of the sources through eq.(4).

In summary, the algorithm starts with initial conditions $W = I$ and $W_l = 0$, $L' \geq l \geq 0$, then updates for $W, \hat{W}$ and W_l are calculated through eqs (10), (11), (12) respectively, matrices D and $W_0{}^0$ are calculated though eqs (16), (17) respectively, matrices $W, \hat{W}$ and W_l are corrected using matrix D as discussed, and finally the new estimates for the sources are calculated using eq.(6). The whole process is iterated until numerical convergence is achieved.

Note: The above derived update rules were based on the stochastic gradient method for training of neural networks. It is, however, the "natural" gradient that gives the optimum update for the system parameters (faster convergence, equivariance property, etc)[1, 3]. The derivation of those update rules will be presented in a future publication, but an interesting lemma is provided in Appendix B.

4. EXPERIMENTAL APPLICATIONS

These methods are currently being applied to two systems of interest. The first is the case of speech sources which are mixed, filtered and delayed. The second is to process EEGs recorded from the surface of auditory cortex of ferrets through the use of a thin microfilm array [20]. In the first case, short time dependencies of the sources are known, so further deconvolution with a pre-whitening scheme is possible. In the second case though, they are unknown, and so further processing depends on the statistics of the filtered sources separated by the above algorithm.

5. APPENDIX A - STOCHASTIC GRADIENT ANALYSIS

The Jacobian J of the network contains all the information on how the input affects the output, by providing all combinations of partial derivatives between each component of the output vector with respect to each component of the input vector. Using the entropy at the output of the system as the cost function to train the network is equivalent, as we showed earlier, to using the expected value of $\ln(|J|)$. In particular:

$$J = \left[\frac{\partial y_i}{\partial x_j}\right]_{ij} = \left[\frac{\partial y_i}{\partial u_i}\frac{\partial u_i}{\partial u_j}\frac{\partial u_j}{\partial x_j}\right]_{ij} = \left[y_i'(u_i)\frac{\partial u_i}{\partial u_j}\frac{\partial u_j}{\partial x_j}\right]_{ij}$$

Here, we note that the partial derivative $\partial u_i / \partial u_j$ should be calculated using equation (6), and it is not just δ_{ij}. That is a result of the extended architecture used that includes versions of the signals for zero delay on the right hand side of the model also. The determinant of the Jacobian can actually be decomposed into the product of the determinants of the weight matrices for zero delay, and the slopes of the nonlinear functions $y_i(u_i)$:

$$\det(J) = |J| = \det(\hat{W}) \cdot \det(W) \cdot \textstyle\prod_{i=1}^N y_i'(u_i)$$

$$\Rightarrow \quad \ln|J| = \ln(|\hat{W}|) + \ln(|W|) + \textstyle\sum_{i=1}^N \ln(y_i'(u_i)) \tag{18}$$

The derivative with respect to a generic matrix V is:

$$\frac{\partial}{\partial V}\ln(|y_i'(u_i)|) = \frac{1}{y_i'(u_i)}\frac{\partial y_i'}{\partial y_i(u_i)}\frac{\partial y_i(u_i)}{\partial u_i}\frac{\partial u_i}{\partial V} = \hat{y}_i(u_i)\frac{\partial u_i}{\partial V}$$

where $\partial u_i / \partial V$ should be calculated from eq.(8). Now we can calculate the various update rules:

$$\begin{aligned}\Delta W \propto \frac{\partial \ln(|J|)}{\partial W} &\simeq [W]^{-T} + \sum_{i=1}^N \hat{y}_i(u_i(k))\frac{\partial u_i(k)}{\partial W} \\ &\simeq [W]^{-T} + diag^1[\hat{W}^T \cdot \hat{\mathbf{y}}(k) \cdot \mathbf{x}^T(k)] \end{aligned}\tag{19}$$

$$\Delta \hat{W} \propto \frac{\partial \ln(|J|)}{\partial \hat{W}} \simeq [\hat{W}]^{-T} + \hat{\mathbf{y}}(k) \cdot \mathbf{x}(k)^T \cdot W^T +$$

$$\hat{\mathbf{y}}(k) \cdot \sum_{l=0}^{L'} \mathbf{u}^T(k-l) \cdot W_l^T \simeq [I + \hat{\mathbf{y}}(k) \cdot \mathbf{u}(k)] \cdot [\hat{W}]^{-T} \tag{20}$$

$$\Delta W_l \propto \frac{\partial \ln(|J|)}{\partial W_l} \simeq \mathrm{diag}^0[\hat{W}^T \cdot \hat{\mathbf{y}}(k) \cdot \mathbf{u}^T(k-l)] \tag{21}$$

since for the general case that $\mathbf{u} = A \cdot B \cdot \mathbf{x}$,

$$\sum_{i=1}^N y_i \frac{\partial u_i}{\partial B} = A^T \cdot \mathbf{y} \cdot \mathbf{x}^T.$$

6. APPENDIX B - NATURAL GRADIENT CONSIDERATIONS

Lemma: The "natural" gradient update for matrices that have and should maintain zero elements can be calculated as if there were no constraints, where at the end the corresponding update elements are set equal to zero.

Proof: Consider the cost function $\phi(\mathbf{w})$ where some of the elements of the vector $\mathbf{w}$ are and have to be kept zero under the new updates. The "natural" descent direction updates $\mathbf{w}$ by $\mathrm{d}\mathbf{w}$, so that $\phi(\mathbf{w}+\mathrm{d}\mathbf{w})$ is minimized when $\mathrm{d}\mathbf{w}$ has a fixed length in the Reimannian space, $\|d\mathbf{w}\|^2 = \varepsilon^2, \varepsilon > 0$, [1]. If we set $d\mathbf{w} = \varepsilon \cdot \mathbf{a}$, equivalently we can search for that $\mathbf{a}$ that minimizes

$$\phi(\mathbf{w} + d\mathbf{w}) = \phi(\mathbf{w}) + \varepsilon \nabla \phi^T(\mathbf{w}) \cdot \mathbf{a}$$

under the constraints:

$$\|\mathbf{a}\|^2 = \sum_{ij} g_{ij} a_i a_j = \mathbf{a} \cdot G \cdot \mathbf{a} = 1 \tag{22}$$

$$\text{and} \qquad \mathbf{a}^T \cdot A \cdot \mathbf{a} = 0 \tag{23}$$

where G is the Riemannian metric tensor, and A is a diagonal matrix with diagonal elements 1 if we require the corresponding a_i to be equal to zero, or 0 otherwise (note $A^2 = A$). In A we have combined all constraints into a single equation which makes it easier to work with the Langrangian method:

$$\frac{\partial}{\partial \mathbf{a}}\left[\mathbf{a}^T \cdot \nabla\phi(\mathbf{w}) - \lambda_1 \mathbf{a}^T \cdot G \cdot \mathbf{a} - \lambda_2 \mathbf{a}^T \cdot A \cdot \mathbf{a}\right] = 0$$

$$\Rightarrow \nabla\phi(\mathbf{w}) = 2\lambda_1 G \cdot \mathbf{a} - 2\lambda_2 \cdot A \cdot \mathbf{a}$$

$$\Rightarrow \mathbf{a} = \frac{1}{2\lambda_1}\left[G + \frac{\lambda_2}{\lambda_1}A\right]^{-1} \cdot \nabla\phi(\mathbf{w}) \quad (24)$$

Applying the Matrix Inversion Lemma below, we get:

$$\mathbf{a} = \frac{1}{2\lambda_1}\left[G^{-1} - \frac{\lambda_2}{\lambda_1}G^{-1}A\left(G^{-1}\frac{\lambda_2}{\lambda_1}A + I\right)^{-1} G^{-1}\right]\nabla\phi(\mathbf{w})$$

$$\Rightarrow \mathbf{a} = \frac{1}{2\lambda_1}\left[G^{-1} - \frac{\lambda_2}{\lambda_1}G^{-1}A\left(G + \frac{\lambda_2}{\lambda_1}A\right)^{-1}\right]\nabla\phi(\mathbf{w}) \quad (25)$$

By eq.(23) we have:

$$\mathbf{a}^T \cdot A \cdot \mathbf{a} = 0 \Rightarrow \mathbf{a}^T \cdot A^2 \cdot \mathbf{a}$$
$$\Rightarrow \mathbf{a}^T \cdot A^T \cdot A \cdot \mathbf{a} = 0$$
$$\Rightarrow \|A \cdot \mathbf{a}\|^2 = 0 \Rightarrow A \cdot \mathbf{a} = 0$$
$$\Rightarrow_{(24)} A\left(G + \frac{\lambda_2}{\lambda_1}A\right)^{-1} \nabla\phi(\mathbf{w})$$

So eq.(25) with the addition of eq.(26) gives:

$$\mathbf{a} = \frac{1}{2\lambda_1}G^{-1} \cdot \nabla\phi(\mathbf{w}) \Rightarrow \mathbf{a} \propto G^{-1} \cdot \nabla\phi(\mathbf{w})$$
$$\Rightarrow d\mathbf{w} \propto G^{-1} \cdot \nabla\phi(\mathbf{w}) \quad (26)$$

which is the natural gradient update for $\mathbf{w}$ if there were no constraints on the values of the $\mathbf{w}$ elements. So, only after we calculate $d\mathbf{w}$ with the above rule, do we zero out the updates that we want to eliminate. QED.

Matrix Inversion Lemma: If A and C are non-singular square matrices respectively, then

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1}$$

7. REFERENCES

[1] S.-I. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

[2] S.-I. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems 8 (NIPS-95)*, pages 757–763, Cambridge MA, 1996. MIT Press.

[3] S.-I. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang. Multichannel blind deconvolution and equalization using the natural gradient. In *Proc. of IEEE International Workshop on Wireless Communication*, pages 101–104, April 1997.

[4] S.-I. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang. Multichannel blind deconvolution using the natural gradient. In *Proc. 1st IEEE Workshop on Signal Processing App. Wireless Comm.*, Paris, France, 1997.

[5] A. J. Bell and T. J. Sejnowski. Fast blind separation based on information theory. In *Proc. Intern. Symp. on Nonlinear Theory and Applications*, Las Vegas, USA, Dec. 1995.

[6] A. J. Bell and T. J. Sejnowski. An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[7] A. J. Bell and T. J. Sejnowski. A non-linear information maximisation algorithm that performs blind separation. In *Advances in Neural Information Processing Systems 7, NIPS-94*, Cambridge MA, 1995. MIT Press.

[8] J.-F. Cardoso and B. Laheld. Equivariant adaptive source separation. *IEEE Trans. on Signal Processing*, 44(12):3017–3030, Dec. 1996.

[9] A. Cichocki, A. Amari, and J. Cao. Blind separation of delayed and convolved signals with self-adaptive learning rate. In *Proc. International Symposium on Nonlinear Theory and its Applications, NOLTA-96*, pages 229–232, Japan, Oct 1996. Research Society on NTA, IEICE.

[10] A. Cichocki, S. Amari, and J. Cao. Neural network models for blind separation of time delayed and convolved signals. *Japanese IEICE Transaction on Fundamentals*, E80-A(9):1595–1603, Sept. 1997.

[11] S. C. Douglas, A. Cichocki, and S.-I. Amari. Multichannel blind separation and deconvolution of sources with arbitrary distributions. In *IEEE Workshop on Neural Networks for Signal Processing, NNSP-97*, pages 436–445, NY, Sept 1997. IEEE Press.

[12] M. Girolami. An alternative perspective on adaptive independent component analysis algorithms. *Neural Computation*, 10(8):2103–2114, 1998.

[13] M. Girolami, A. Cichocki, and S.-I. Amari. A common neural network model for exploratory data analysis and independent component analysis. *IEEE Transactions on Neural Networks*, 9(6):1495–1501, 1998.

[14] Y. Guo, F. Sattar, and C. Koh. Blind separation of temporomandibular joint sound signals. In *Proceedings ICASSP*, 1999.

[15] T.-P. Jung, C. Humphries, T.-W. Lee, S. Makeig, M. J. McKeown, V. Iragui, and T. J. Sejnowski. Extended ica removes artifacts from electroencephalographic recordings. In *Advances in Neural Information Processing Systems 10, NIPS-98*, 1998.

[16] K. H. Knuth. Difficulties applying recent blind source separation techniques to EEG and MEG. In *Proc. Maximum Entropy and Bayesian Methods workshop,MaxEnt97*, pages 209–222, Boise Idaho, Aug. 1999. Kluwer, Dordrecht.

[17] T.-W. Lee, A. J. Bell, and R. H. Lambert. Blind separation of delayed and convolved sources. In *Advances in Neural Information Processing Systems 9, NIPS-97*, pages 758–764. MIT Press, 1997.

[18] T.-W. Lee, M. Girolami, and T. J. Sejnowski. Independent component analysis using and extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation*, 11(2):409–433, 1999.

[19] T.-W. Lee and R. Orglmeister. A contextual blind separation of delayed and convolved sources. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processings, ICASSP-97*, pages 1199–1203, April 1997.

[20] A. L. Owens, T. Denison, H. Versnel, M. R. M., and S. A. Shamma. Multi-electrode array for measuring evoked potentials from surface of ferret primary auditory cortex. *Journal of Neuroscience Methods*, 58:209–220, 1995.

[21] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1965.

[22] D.-T. Pham and P. Garrat. Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Trans. on Signal Processing*, 45(7):1712–1725, 1997.

[23] P. Smaragdis. Blind separation of convolved mixtures in the frequency domain. In *Proc. International Workshop on Independence & Artificial Neural Networks*, University of La Laguna, Tenerife, Spain, Feb. 9–10 1998.

[24] K. Torkkola. Blind separation of convoluted sources based on information maximization. In *Proc. of IEEE Workshop on Neural Networks for Signal Processing NNSP-96*, Kyoto,Japan, Sept 4-6 1996.

[25] K. Torkkola. Blind separation of delayed sources based on information maximization. In *Proc. IEEE ICASSP*, Atlanta, USA, May 7-10 1996.

[26] K. Torkkola. IIR filters for blind deconvolution using information maximization. In *Proc. of Workshop on Blind Signal Processing, NIPS-96*, Snowmass, CO, Dec 7 1996.

[27] K. Torkkola. Blind seperation for audio signals - are we there yet? In *Proc. Workshop on Independent Components Analysis and Blind Signal Separation*, Aussois, France, Jan. 11-15 1999.

[28] L. Zhang, S. Amari, and A. Cichocki. Natural gradient approach to blind separation of over- and undercomplete mixtures. In *Proc. ICA99*, pages 455–460, Aussois, France, Jan 1999.